

OCTOBER
2013



Vol. 23 N° 7 £5

Archive

*The subscription magazine for users of the
Acorn Archimedes and its RISC os successors*

Newsbriefs ...	Jim Nagel ... 3
<i>London show preview; Disc Knight update; Pluto is back; new clock for old machines; hopes for Impression X</i>	
Portsmouth by Twitter ...	Martin Hansen ... 8
Wakefield show 2013 ...	Nick Mason ... 10
	Vince Hudd ... 12
Another slice of Pi ...	John Schild ... 16
<i>Raspberry Pi takes over as main machine</i>	
Greetings from Photodesk ...	John Schild ... 22
<i>Making a Christmas card on the Raspberry Pi</i>	
Coming back to Ro ...	Malcolm Hussain-Gambles ... 26
<i>Raspberry Pi rekindles love for Riscos</i>	
ArminiX: new era in Ro audio ...	Jim Lesurf ... 28
Illusions – jpeg compression ...	Gerald Fitton ... 34
<i>Lena: 'the First Lady of the internet'</i>	
Secret life of algorithms ...	Nicholas Cutler ... 39
<i>(9) Jpeg compression</i>	
Mac Matters ...	Mark Stephens ... 46
<i>Running RPCemu on both Mac and Linux</i>	
PC Bits: at the support desk ...	Bimal Jangra ... 48
Compiling with GCC (2) ...	Rob Johnson ... 52
Charm – a practical guide ...	Pete Nowosad ... 58
<i>Part 2: Data</i>	
More ednotes ...	Jim Nagel ... 62
<i>Armini initiation; a Movember jape for the London show</i>	
Archive forum and files to download ... 2	

ednotes *(And more on page 62)*

- See you at the London Risc os show on October 26! This issue of Archive has reports of previous shows, at Portsmouth and Wakefield, which will flag up some of the exhibits you will want to check out at London. (See pages 8–15.)
- Here's a first in my time as editor: two Archive authors independently sent articles on the same topic. Nicholas Cutler and Gerald Fitton both chose to write about the fascinatingly clever Jpeg method of compressing photos. They come at it from quite different angles, as you'll see on pages 34–45.
- Did you receive an email from Archive in the past 10 days? (It was a reminder about the London show.) If not, then please amend the membership database by sending an email to updates@archivemag.co.uk stating your postcode or membership number as positive ID. Any other message, though, should go to the normal address.
- Thanks to Martin Bazley, James Harper, Chris Newman and George Pearce for helping to proofread this issue. Volunteers for next time, please contact me. —jim

Cover – *The immense Celtic cross greeting visitors as they cross the Tamar is Cornwall's echo to the Angel of the North. John Schild tells more about his photo on page 22.*

Archive was founded in 1987 by Paul Beverley in support of the Acorn Archimedes computer and its successors using the Risc os operating system. It is entirely produced with Risc os software, notably Impression Publisher and Artworks. Body text is 9½pt Plantin; headings are EFF Eric (Gill Sans). Litho this time by Bishops Printers, Portsmouth. Wordcount this issue: 27,360.

Editor: **Jim Nagel** jim@archivemag.co.uk

SUBSCRIPTIONS – £50 in the UK for 12 issues, £56 to the rest of Europe, £62 to rest of the world

SIX-ISSUE SUBSCRIPTIONS – £30 in the UK, £35 to the rest of Europe, £38 to the rest of the world



*All the past
22 volumes on
CD or DVD:
see page 62*

Payment methods: cheque to “Abbey Press” at address on back cover, or direct bank transfer with an email to confirm details – sortcode **08-92-50** (Co-op, Skelmersdale), account **7072 6592** (Abbey Press). Google Checkout is no longer available, but see www.archivemag.co.uk (Prices page) for other routes.

Archive files to download free



Several articles in this issue have files to go with them, as flagged by this graphic in the text. Go to archivemag.co.uk and click on Files.

If you are not online and would like the files on a floppy, just contact the editor.

➤ **An index to your copies of Archive** – Cumulative Arcscan datafiles are also on the Files page of the website. Or click Search for Kris Adcock's online service.

➤ **All web links** printed in this issue are also on the website in clickable form: see URLs.

Join the online forum – Archive-online is a friendly mailinglist where Archive members help one another with problems. It's gently moderated by Keith Johnson (him of the East Fleggs). To join, email forum@archivemag.co.uk with subject Subscribe.

Newsbriefs **Jim Nagel**

London show set for October 26

The London Risc os show for 2013 takes place on Saturday October 26. The show is the fifth organized by Rougol, the Risc os User Group of London.

A new exhibitor this year is Richard Keefe with Impression X, demonstrating his progress at updating the mammoth mass of assembly-language code to let the classic Risc os DTP application run natively on modern machines. He hopes to have at least a new Jpeg-import feature ready for release at the show. *[More on page 6.]*

Risc os Open Ltd will have version 5.20 of the operating system running on all hardware from the new Arm incarnations back to RiscPCs and A7000s.



Raspberry Pi enthusiasts will show off all sorts of clever tricks they have contrived for the tiny device.

Aemulor will be at the London show again, enabling classic 26-bit applications to run on new 32-bit-only hardware, including the Pi.

CJE will show its ready-to-go Panda Ro and Raspberry Ro machines along with its usual extensive array of accessories.

R-Comp's Arm-in-One integrates the computer into a high-definition IPS screen as a tidy space-saver for £399, and the screen has extra inputs for a DVD player or another computer. The ArminiX – also built around the Pandaboard – now comes with latest hardware and OS pre-installed. The Pi Plus support scheme offers 30M of enhancements for Raspberry Pi Risc os

users in a single drag. On the software front, R-Comp will have an update of Messenger Pro that copes better with emails containing non-English alphabets.

Plingstore 1.0 will be new at the show. It's the Risc os reply to Google Play and Apple Store: an online place to find both free and commercial apps from many developers.

Techwriter and Easiwriter's new version, 9.1, will be on sale by the developer, Martin Würthner. *[More on page 6.]*

Phoebe is another attraction: the only working prototype of the RiscPC successor that Acorn cancelled in 1998. The Centre for Computing

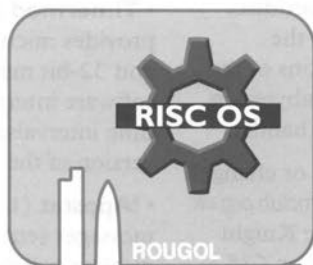
History is bringing her to the show in her bright yellow case.

The show's charity stand can accept any Acorn or Risc os hardware or software, but not CRT monitors or x86 PCs. The stand is in aid of Combat Stress, a charity to help veterans with mental-health problems.

The London show is open from 11am to 5pm; entry is £5 (under-16s free).

The location is again the St Giles Hotel in Feltham, on the western edge of London just south of Heathrow. It has good bus links and trains from London Bridge and Waterloo every 10 minutes; Feltham station is ten minutes' walk from the hotel. For satnavs, the postcode is TW14 9AD; if the hotel carpark is full there are others nearby. Updates, full show details and travel hints are at riscoslondonshow.co.uk.

Reports of the Portsmouth and Wakefield shows on pages 8–15 highlight some of the things to look out for at London.



• **Blue eyes shut** – The venue for Rougol's monthly meetings, the Blue-Eyed Maid pub in Southwark, is closed for six weeks. Check rougol.jellybaby.net, for the location of the November and December meetings. "We have an unbroken run of monthly meetings going back 10 years," said the chairman, Bryan Hogan.

Sir Druck to the rescue



A regeneration of **Disc Knight**, the essential utility by David Ruck for repairing a harddisc, adds support for Risc os 5.2 and the Raspberry Pi's 5.19. These versions of the OS remove the old limit of 4 gigabytes on the size of files that Filecore can handle.

To upgrade to Disc Knight 1.50 or change your registered address, go to armclub.org.uk and follow the links. Buying Disc Knight afresh costs £10 for a single user or £25 for multiple machines at the same site.

One of the best features of Disc Knight is personal email support from the author when necessary. "I encourage everyone to upgrade to the latest version as it makes support much easier," he said.

For older machines, you must keep a copy of Disc Knight on a floppy in case your harddrive fails. Newer machines have no

floppy drive but have USB support in rom, so keep Disc Knight on a memory stick that is formatted as *Filecore* – do not depend on anything, such as Fat32FS or networking, that runs from !Boot on the drive you need to rescue. "Always try it out before you need it!" said Druck.

Also on the Arm Club site are free new versions of three small programs. These are updates to cope with the Raspberry Pi, and run on all recent versions of Risc os.

• **Timermod** (the new version is 4.17) provides microsecond timing on 26-bit and 32-bit machines. Star-commands and software interrupts can be used to measure time intervals; there is also a microsecond version of the monotonic timer.

• **!Appstat** (1.24) monitors events and messages sent to all running applications, displays call counts and time spent in applications to microsecond accuracy.

• **!CPUload** (3.23) is an improved CPU usage monitor. It supports multiple windows and displays the load carried by the entire machine or selected tasks.

Beware attempted fraud

A warning from Michael Harding, who invested in shares back in 1998 to keep Acorn going. Like many other Archive readers, he still holds some of his shares.

"I had a fishy (phishy?) phone call from Eclipse Options Ltd offering me stock options in Arm, supposedly as a reward by Arm to its longstanding investors." Names like Morgan Stanley were bandied about, he said. Of course the cold-caller said the options have to be taken up very quickly.

"I rang Equiniti (which pays out the Arm dividends), and an advisor told me a fraudulent offer of share options is doing the rounds. Arm put information on its



First Drucklet
Alexander William Michael was born to David and Louis (Smith) Ruck on September 29.

website warning about it, but obviously can't close the scam down instantly. So beware!

"Incidentally, Equiniti's young advisor had no concept of what Arm is, imagining it was some sort of software or operating system!"

The so-called "boiler room" scams offering to buy Arm shares at inflated prices have increased in recent years, Arm's *Warning to shareholders* page says. "They promise high returns, but those who invest usually end up losing their money."

Pluto pops out of eclipse



Pluto, the still-popular client for email and newsgroups, has been upgraded to run on new Arm hardware. Jonathan Duddington wrote Pluto in 1997 as a commercial product but has been silent for a few years. As an Easter surprise he released version 3.06 and announced that Pluto is now free and open-source. His only changes (sourceforge.net/projects/plutoemail) were to suit it for new Arm hardware such as Raspberry Pi and Armini.

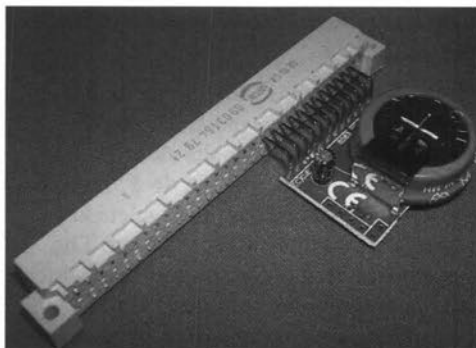
Devoted Pluto users immediately got to work. Rob Sprowson put out two updates in summer and collaborated with Martin Avison to release 3.09 in September: www.avisoft.f9.co.uk/Pluto.htm. It comes with an updated Stronghelp manual.



Since 2005 Jonathan, in Coventry, has devoted his main development efforts to his compact "**eSpeak**" program, which began life in 1995 as a Risc os application called !Speak. Its purpose is to read text aloud, and it serves many blind users – it's the default speech synthesizer in the Windows program NVDA (non-visual desktop access) and the Linux counterpart called

Orca; it has also been used by Google Translate.

Espeak is included within Pluto or can be downloaded on its own from espeak.sourceforge.net. It comes with a selection of languages, and more can be obtained from 51 listed at the Sourceforge site. Espeak can be used to read text from other Risc os software such as StrongEd, Textease or Messenger Pro.



If the hardware clock in an Acorn-era machine is damaged, CJE's podulette can replace it.

New clock for old machines

CJE Micros has brought out a replacement clock module for the RiscPC and older Acorn machines back to the A3000.

This will be of interest to users who thought to remove the battery before putting a superseded computer into storage – knowing that a battery leak can cause irreparable damage to the motherboard – only to find that after months without power the onboard clock may refuse to start again.

The new CJE clock comes on a podule for the A5000, A7000 and RiscPC and a minipodule for the A3000–4000 series. Price is £26 plus £2 carriage. Fitting the podule is simple; disabling the old clock is done just by snipping its legs with fine

side-cutters (which CJE can hire if the customer doesn't already have a pair), as shown in photographic instructions at cjemicros.co.uk/rpctc.

No additional software is required; the operating system will automatically recognize the new clock and battery-backed "C-mos" memory.

"We plan to show how easy it is to do by letting customers do it on our stand at the next few Risc os shows," said Chris Evans of CJE Micros.

(If the battery has leaked while in storage, it could be that only the clock chip has been damaged, since it sits close to the battery, Chris said. To test, do a "Delete-power-on" – hold down the Delete key while you switch on the computer. The system will do a power-on self-test, which may take up to two minutes. "If the computer then gets to a desktop and you can access your drives (which of course might involve restoring some settings reset by the Delete-power-on), that suggests the system is working other than the clock.")

Impression X aims for London

An interim release of Impression X – with bugs fixed and a new long-overdue ability to import Jpeg pictures – just might appear at the London show on October 26.

Richard Keefe, the new developer of the classic Risc os publishing application, told Archive he is concentrating on these two aspects "as I have realized that the work of fully 32-bitting Impression is going to take longer". Meanwhile Aemulor is available for all the new Arm hardware, including the Raspberry Pi, to run existing 26-bit versions of Impression.

The Jpeg import will make use of the free Artworks rendering module. "Due to the way Impression renders the files for

printing, it is quite complicated," Richard said. This new ability will keep Jpeg pictures in their native format and small file size, unlike the old Transjpeg module from Computer Concepts, which changed Jpegs to sprites within the Impression file.

Richard obtained the source code of Impression over a year ago (more detail in *Archive 23:6*) from Paul Reuvers of Xat

Technology, who gained the licence from Computer Concepts at the same time as Martin Würthner took over Artworks from CC.

Richard said his agreement allows him to sell new copies of the existing CC program –



IMPRESSION
Style

Mailinglists are running for discussion of Impression X. Users of Publisher and Style should register their interest by sending a blank email with subject "subscribe" to impression@freelists.org or impression-bugs@freelists.org

but it is hard-coded for floppy discs, so he will have to devise a way of packaging it for new machines that lack floppy drives.

More software updates

• **Easiwriter** and **Techwriter** version 9.1 offers greatly improved export as web pages using HTML4 and CSS2. Browsers show a result that looks much more like the original document: accurate font size, spacing, indents, background colours, borders and so on.

Version 9.1 also copes better with importing Microsoft Word documents that were saved in .docx format. It imports background shading of paragraphs and tables. In tables, it deals with multiple paragraphs in a single cell, column widths and floating table alignment.

One user, Stewart Brookes, commented on a further new feature: "It's amazing how much usefulness is added by an apparently small enhancement in the form of the new visibility option for headers and footers."

Martin Würthner, the maintainer, took advance orders for 9.1 at the Wakefield show but held it back till summer to add further features. mw-software.com

• **Netsurf's** daily build number was just short of 1,400 at press time: the team of volunteer coders is working on Javascript support for the free web-browser. Back in the spring, the handling of multi-line text fields in input forms – the ground of many crashes in the past – had a thorough overhaul, thanks to Michael Drake; undo and redo (on the usual Ro keys F8 and F9) now work too. In October the Risc os mouse coding and hotlist had a major makeover by Steve Fryatt to make it easier to maintain. Download the latest development build from netsurf-browser.org

• **Risclua** – the Lua programming language for use in Risc os – reached version 5.70 in August. It has caught up to the latest version 5.2.2 of the international language and uses the latest libraries. In the Ro front end, Gavin Wraith and Michael Gerbracht have cleaned up the wimp and toolbox libraries and added new update operators. Download it from www.wraith.plus.com/lua along with examples and documentation. Lua is particularly adept at matching patterns and catching data within text.

• The **Virtual Acorn booklets** have been rejigged by the authors at TOMS in Yorkshire to display on tablets, such as the Android one sold by R-Comp, and on the smaller screen of the Kindle e-reader. You can also print your own copy as an A5

booklet. It's free to download from virtualacorn.co.uk/support/vrpinuse. Virtual Acorn marked a quiet 10th anniversary last month although its RiscPC emulator that runs on Windows and Mac caused loud ructions in the Risc os world when it first came out in September 2003.

• **Stampit**, the little utility by Ian Hamilton to change the date on a file, has a slight update. Version 2.09 adds a "zero time" button to set hours, minutes and seconds all to 0 – which your editor finds useful to signal that he has stamped an after-the-fact date. hamiltoni.pwp.blueyonder.co.uk

Ro in hi-fi mag ...

The latest issue of *Hi-Fi News* (cover date "Yearbook 2013", on newsstands now) has a one-page feature by Jim Lesurf. "I made the ArminiX my 'highlight of the year' in my opinion column," he told Archive. "They also printed some photos I took, including one showing Risc os and DigitalCD playing a Flac file at 96K sample rate." [See Jim's article "ArminiX – a new era in Ro audio" on page 28 in this issue of Archive.]

... and Rool in the Guardian

A feature in the *Guardian* on September 18 questions Apple's use of 64-bit chips in its much-vaunted new Iphone 5: "useful, necessary or just for show?" The authors are "engineers who have worked with Arm technology since the 1990s" – who turn out to be Andrew Hodgkinson, Steve Revill and Ben Avison, the crew at Risc os Open Ltd. The article is online: is.gd/AYg2jc

Southwest show on March 1

Another Southwest show is to happen, thanks to R-Comp and Orpheus Internet, the sponsors for the past few years. As usual it's at the Webbington Hotel overlooking the M5 south of Bristol – on March 1 in 2014. www.riscos-swshow.co.uk

Portsmouth by Twitter

Martin Hansen @RISCOScode

🐦 **Saturday September 28, 09:14** Welcome to Riscoscode's coverage of the first Risc os Portsmouth show. If you're in Portsmouth today between 11am and 5pm drop in to say hi.

🐦 **09:21** In case you are nearby: the show is on the second floor of the Innovation Space Building, 1 Hampshire Terrace, PO1 2QF.

🐦 **09:25** Right now the first exhibitors' cars are being unloaded and the first stalls being set up. I need to do likewise now with Riscoscode's stall.

🐦 **09:57** First sale of the day, before the show even opens, goes to CJE Micros: an email from Sweden ordering a second-hand Iyonix.

🐦 **11:02** Panic stations at the show: No sign of R-Comp, one of the show's big hitters. Otherwise all exhibitors seem ready for the start.

🐦 **11:20** Panic over: R-Comp has arrived, having explored all of Portsmouth on the way. Time for a Risc os GPS system perhaps.

🐦 **12:03** Exhibition hall is pleasingly full of people and there's a lovely chilled and relaxed atmosphere.

🐦 **12:09** Been talking with Richard Keefe about the work he's doing on rescuing the Risc os killer app, Impression. Great to hear progress is being made. **12:14** Impression is a wordprocessor written in old 26-bit Arm assembler code. Richard is making it 32-bit-neutral so it'll run natively on Iyonix and Pi.

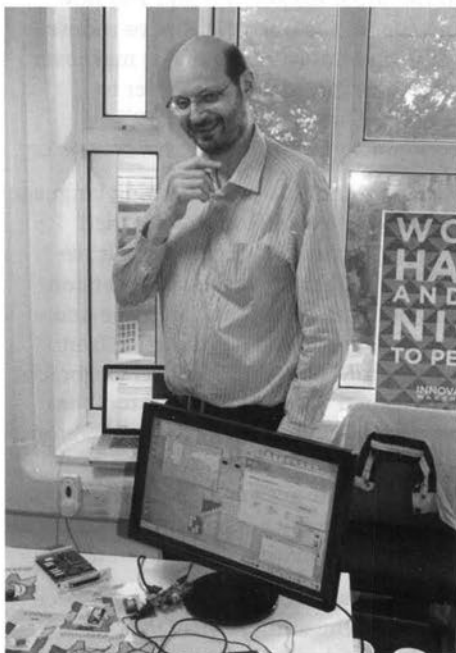
🐦 **12:21** Breaking: Vince Hudd has announced a website to allow Risc os

enthusiasts to vote for the best and worst of Risc os: riscosawards.co.uk

🐦 **12:24** Bryan Hogan and several other members of the Risc os User Group of London (Rougol) have just arrived. The London Show in a month's time is shaping up well.

🐦 **12:27** Been admiring Chris Hall's book *Getting Started with RISC os on the Raspberry Pi*. It's very smartly produced and only £4.50 at the show.

🐦 **13:14** *Drag'n'Drop's* editor [Christopher Dewhurst] has a new issue of the Risc os PDF magazine in the pipeline. Tentatively scheduled for release on October 12.



Ben Avison of RISC os Open Ltd was kept busy selling Rool pendrives and CDs containing Ro 5. Wearing a second hat, as Piccolo Systems, he had his own software such as Clonedisc on offer.

13:17 CJE Micros is selling a camera board for the Raspberry Pi. If I could run it using Risc os I'd have bought it. Alas, no drivers for it.

13:20 Ron Biscoe says the Riscocode display of a battery-powered Pi and 22-inch monitor would be of use to folks in caravans and motor homes.

13:23 Spotted Gavin Wraith across the hall putting up posters to advertise the programming language Lua for Risc os. Am off to chat to him.

14:36 Bought a violently red case for the Riscocode Raspberry Pi and also a realtime clock from CJE Micros. (Yes, I checked that clock fits in case).

14:40 Risc os Open Ltd is doing good business. Ben Avison is doing a good job dealing with customers, fielding their questions, taking their cash.

14:42 A couple of customers at my Mathmagical stand have spotted that the computers there are running a new RPi desktop theme – which we hope to release “real soon”.

15:46 Talking with Andy Moyler, who is responsible for this show at Innovation Warehouse being free for both visitors and exhibitors. **15:48** He's passionate about tech in general and Risc os in particular, and wants to get it in front of schoolkids and university students. **15:50** His challenging question: *What can the Risc os community help him place in front of youngsters that will inspire and enthuse them for Risc os?* **15:52** This is not just an idle question; this is the whole purpose of his job here at Innovation Warehouse.

15:56 Just did fly-by of the Organizer stand, where the version on display looks so very much better than my version from three years ago.

15:59 Am with R-Comp's Andrew Rawnsley – I want to quiz him about Plingstore. This is fast becoming the Ro version of Apple's App store. **16:02** Very impressed with !Store. A slick way to download Risc os software. It's great R-Comp is so focused on a good user experience.

16:04 Most frequently downloaded item on !Store since it opened last October is the (free) Atari ST emulator, Hatari (more than 1,000 downloads).

16:18 Steve Drain is showing off a super-slim Raspberry Pi plugged into the back of a Motorola laptop case. It's a one-off build. Very cool.

17:24 Stewart Pearce, another member of the Innovation Warehouse staff, has been enthusing to me about what he's seen at the show today.

17:24 Last visitor has left, the show is over. Now to pack up. Hope you've enjoyed today's Twitter coverage of the Risc os Portsmouth show 2013.

Martin Hansen mhh@shrewsbury.org.uk



Martin teaches at Shrewsbury School and exhibits his Mathmagical software at shows and floats a Pi-powered boat, and in his spare time keeps the Riscocode.com website up to date.

Club innovation

Chris Newman, who lives in Portsmouth, adds that the Southampton Risc os user group had a stand at the show “displaying all sorts of machines and bits and pieces – a new venture for us”.

Impresario's verdict

It was Risc os Open's first go at organizing a show. Steve Revill said: “We're feeling inspired to have another pop next year – picking a date a little further away from other events on the Risc os calendar.”

The Wakefield show 2013

Ednote: I had to cancel my trip to Wakefield on April 20 (my wife broke her arm three days before the show) and in the flurry neglected to noble somebody to write up a report for Archive. So I sought snippets from various people to collate into a report.

Nick Mason posted in the newsgroup comp.sys.acorn.misc just before the show hoping people would bring along BBC Master software to complete his collection. I emailed him to ask how things went.

Nick Mason

Unfortunately, only a single person dropped off a single small cardboard box of BBC Master Compact magazine discs, so I didn't really end up with any of what I wanted. Nevertheless, there were a couple of interesting things on those discs and, on a plus note, I was able to buy quite a few Risc os packages from the charity stand, so the Wakefield Hospice ended up with a decent contribution.

I'm mostly interested in educational software (although any games are always welcome). Now that I'm the proud father of a beautiful baby daughter, when the time is right, I want to show her some of the things that I grew up with. Whether she'll be in the slightest bit interested though, is another matter.

It was a shame that you couldn't make it to the show, Jim – but Vince Hudd had put a clear message on the empty Archive stand apologizing for your absence. I hope your wife has a speedy recovery. *[Yes, thanks: pretty well back to normal now. She was off for eight weeks and returned to school just in time for end-of-term parties. —Ed.]*

Best turnout in years

On to the show then. The turnout was excellent; I've not seen a Risc os show as packed for many years. Another member and I discussed why this was so – perhaps the resurgence of new Ro-compatible hardware and the Raspberry Pi.

Risc os Open Ltd, familiarly known as Rool, had a good stand selling various things including **Risc os for the Raspberry Pi**. This was very helpful, as many SD cards don't work properly with the Pi Risc os port, so having a ready-to-go SD card with assured compatibility was a great idea.



Even better though was an SD card called **Nut Pi** – a decent collection of commercial software from Acornland on an SD card for the Pi, for £35 plus Vat. This contained gems such as Impact, Datapower, Organizer, Safestore and Photodesk, not to mention Rool's DDE (Desktop Development Environment) and was excellent value for money. At first I wondered how on earth they could do this – then, I thought: I suspect they're old versions, to introduce users to the various applications. That provides the Risc os software vendors with a whole new audience, especially with the Pi. I think the idea might be that having tried these, the new user might want to upgrade to the latest versions, thus getting the vendors some much-needed sales. Needless to say, I snapped one right up.

Aemulor Pi was unveiled at the show by its developer, Adrian Lees. It was for sale, even though he still had tweaks to do and you couldn't take it home from the show. I paid my £18, and the email came early the following morning with the relevant details



Nut Pi



The dormer windows are the trademark background for the Wakefield show: here on the top floor of the Cedar Court Hotel 17 years ago Acorn unveiled the StrongArm RiscPC and put Wakefield on the map. (Photo by Kevin Urquhart in the Wakefield club magazine.)

attached to download the software from buyit.spellings.net. Its purpose, of course, is to enable legacy 26-bit software such as Impression run on new fully 32-bit hardware. The first version of Aemulor came out for the Iyonix.

I find the Risc os SD card in my Raspberry Pi very nice indeed, fast and pleasant to use. Linux (about which I know plenty) feels very slow in comparison; Ro positively races along.

Theatre packed too

I didn't sit in on many presentations, though I popped my head around the door a few times. Many of the sessions were packed, especially the Rool presentation. Similarly it was standing-room-only for Martin Wuerthner's demonstration of Artworks and Techwriter.

As well, I spoke to Ian Wolstenholme, better known as the Beebmaster, who was very friendly and knowledgeable. He told me that Retroclinic [Mark Haysman] had just been and gone – shame I didn't meet him, because I bought my Master Compact from him via Ebay.

Pi and a webcam

CJE Micros had an array of interesting things for the Pi. I bought a new Raspberry Pi Model B (512M) from CJE, and two cases.

The stand had a nice array of **tiny LCD screens** suitable for the Pi at various price points. I bought the cheapest one at just £28. It looks kind of retro and is ideal for the Pi. My old Pi will be going into my baby daughter's room to stream nonstop music and, I hope, power a webcam so we can keep an eye on her antics when we're out of the room. The small screen will be handy for any maintenance I need to do on the Pi in there – although it is tiny and the definition isn't great, it's fine for small embedded applications, which is just what I wanted.

I also bought a copy of **ProCAD+** through CJE – the application by David Snell for computer-aided design, which can even import Ordnance Survey maps. *[It now runs on new Arm hardware such as the Pi. —Ed.]* I've been looking at this for years, but couldn't really justify the price for

something that is just a passing interest. The special Wakefield price did the trick and I bought a copy – it's absolutely brilliant, very easy to use. And it came with a ring-bound manual that takes you through the various options, which I like. I much prefer printed documentation – when I have a spare minute, I can just pick it up, dive in, and learn something else new about the software. I do wonder why the Windows version of ProCAD is so much more expensive: £153 against £75. Although now I like the Ro version so much, I might be tempted with the Windows version one day.

Relative newcomer to Risc os

It was nice to see so much new Risc os hardware – I hope the Pi attracts more users to the platform. I'm relatively new to Ro myself. I started on it in about 2005 after reading about the A9Home (which I

now have), although I always liked the idea of a rom-based operating system – fast and indestructible (I'm conveniently ignoring the softload versions).

I've used more platforms and graphical user interfaces than most people, but I like Ro most of all – it's very logical and, more importantly, convenient to use. Whenever I have to dig out a menu on Windows several times in succession to tick multiple options, I end up grumbling something like "Grr... if this was Risc os, I'd only need to click with Adjust" and so on.

Nicholas Mason njsmason@googlemail.com



Nick likes finding innovative and elegant solutions to complex programming problems. When he's not busy writing software, he can be found walking or running, in and around his local village of Boothstown, Worsley.

The following is adapted from his Wakefield report at riscository.co.uk, which Vince edits.

Vince Hudd

This year's Wakefield show was the 18th, and again took place at the Cedar Court Hotel, at junction 39 of the M1 – the same venue as for the last few years and where the first show was held all those years ago.

As usual, I was there to demonstrate and discuss my products, and also – in theory – to get a gist of what was happening so that I could report about it on *Riscository*. However, I barely left my stand during the entire show – still tired after an extra long drive the day before, and much busier than usual on my stand.

Numbers up

The show seemed to have attracted more visitors than in recent years. I commented about this to a number of people during

the day, observing that plenty of people were still wandering the show even when the theatre presentations were underway. At the end, while everyone was packing things away, Chris Hughes of the Wakefield Risc os Computer Club (Wrocc) – the show's organizers – confirmed that attendance was up on previous years.

He has since told me that attendance has been on the increase for the last two years, and that this year had 160 visitors compared to around 140 last year – a trend I'm sure we all hope continues, and for all the Risc os shows, not just Wakefield.

The Wrocc stand was, as usual, the first stand visitors saw on entering. The club is now 30 years old, and – opting for Pi instead of cake – was giving visitors and exhibitors the opportunity to enter a draw to win a Raspberry Pi in a clear case. At

the end of the show, Martin Würthner pulled the winning ticket from the tombola, and the Raspberry Pi went to Richard Brown of Orpheus Internet.

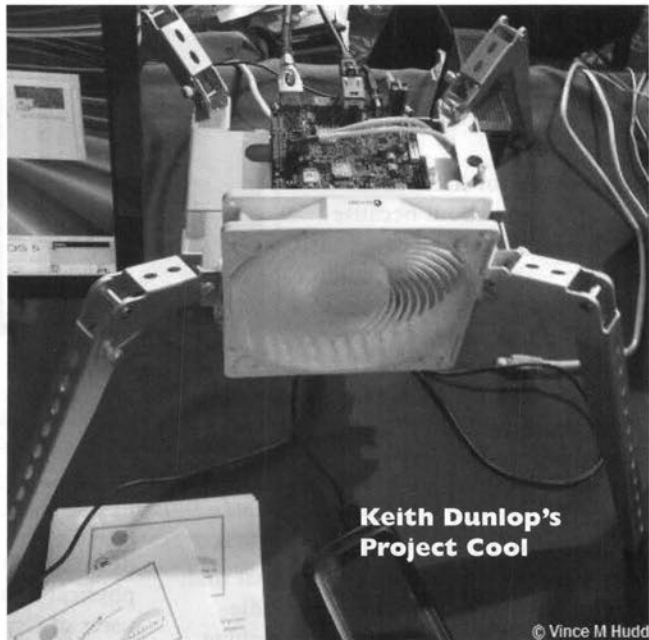
Also available from the Wakefield club's stand was the third edition of the **Wakefield Back Catalogue**, a CD containing WROCC newsletters going all the way back to the very first one, along with a slideshow from the 2008 show, and a video of the 1998 show.

The club's immediate neighbour was the charity stand. This year the total raised for the Wakefield Hospice – by both the charity stand and Steve Fryatt, whose show proceeds also go to the charity – was £810.

Past and future

Describing itself as “a community dedicated to encouraging and supporting the development of brand-new games for ‘retro’ computing platforms” as well as a publishing label for Acorn computers, Retro Software's stand provided new software for older platforms. Among the goodies on sale this year was the game *Repton – the Lost Realms*.

At the Beebmaster stand, Ian Wolstenholme showed off a variety of items from his collection that hold a particular, personal significance to him in the development of the Beebmaster website. Ian had also brought along a selection of goodies to sell from Beebshop, Beebmaster's sister site, through which he



**Keith Dunlop's
Project Cool**

© Vince M Hudd

On the Usable Range stand, Keith Dunlop was showing off Project Cool: a Pandaboard mounted on four struts, with a large fan attached to the front. The fan helps to keep the temperature down – Keith usually overlocks his Pandaboard from its usual 1.2 to 1.5GHz.

The whole thing looked as though it should be walking around, and has the honour of being the only photographs I took this year.

sells spares and upgrades for BBC Micro enthusiasts.

Bridging the old–new gap, the next stand was Jason Flynn's. Jason was demonstrating the latest USB Econet network card, through which modern computers equipped with USB can connect to legacy Acorn Econet networks. Also being demonstrated on this stand was SoftMDFS by Jonathan Harston, running on a Raspberry Pi, serving files onto the show Econet.

Two new computers at R-Comp

One of the busiest stands of the show had to be R-Comp's, where Andrew Rawnsley

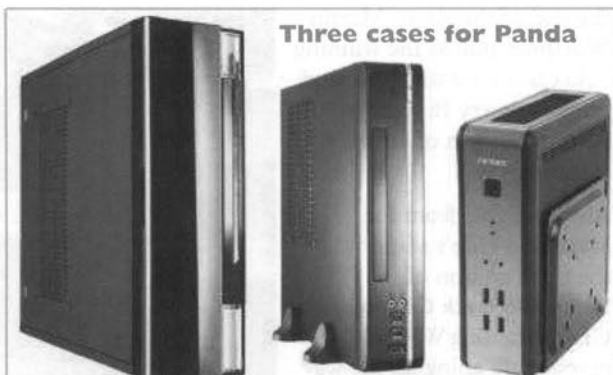
and team appeared to be inundated with visitors throughout the day. R-Comp launched two new computer systems.

Arcimedes (or Arcie to its friends) came about because Andrew discovered a case called the Arc – and R-Comp couldn't resist producing a computer in a case with that name. In a press release a few days before the show, he explained that his first inclination would have been to design a pure-Arm system, but the size of current boards on which Risc os can run – the Pandaboard, Beagleboard and Raspberry Pi – rather dictates against that; the Arc chassis suits a small or middle-sized tower. This is, therefore, a machine which provides a Risc os experience – via emulation – on either Windows XP or Windows 7.

The other machine, the **Armicro**, brought to Wakefield in prototype form, harks back to the old BBC Micro and A3000 designs: it's a "single box" computer, with a keyboard and touchpad incorporated into the chassis. This computer, unlike the Arcie, is pure Arm: it is powered by a "modified Raspberry Pi setup," with a decent amount of high-speed storage. The only thing a user needs to add is a suitable monitor with DVI or HDMI connection.

Sweet internet speeds

Orpheus Internet had the next stand, where Richard Brown was talking about the services offered by the Risc os-friendly ISP (and, more importantly to some people, giving away sweets). All of Orpheus Internet's **fibre-to-the-cabinet** packages now offer a maximum speed of 80Mbps



*CJE increased the specifications and reduced the prices of its **Panda Ro Pro** computers, which are based on the Pandaboard ES and available in a choice of three cases (pandaro.co.uk).*

The smallest case measures 222×78×212mm. The larger two can hold a DVD drive; the middle one has an internal power supply.

down and 20Mbps up, subject to the usual physical limitations (line length, etc). For users of Orpheus hosting, Richard's newest server provides a control-panel interface for administering the service.

At my Soft Rock Software stand next door I was annoyingly just out of reach of Richard's sweets – he had them on the wrong end of his table. As usual, I was giving my usual excuses for not having finished the Webchange manual, among the many other things I haven't done



Visitors to my stand could purchase a new edition of Chris Hall's book, **Getting started with Riscos on the Raspberry Pi**, now expanded to include some Basic programming

instruction, including a complete type-in listing of his Cat application, which presents a disc catalogue in the form of a tree diagram. Chris was on hand for much of the day to talk about the book.

Network Organizer

Martin Avison and Nigel Willmott launched version 2.20 of **Organizer**, the electronic diary and addressbook. It has new features that allow Organizer to be used on several workstations on a network, as discussed at the Southwest show earlier this year.

Christopher Dewhurst on his stand was promoting **Drag'n'Drop**, the PDF magazine that he took over from Paul Stewart late last year. Chris was selling a compilation of all 15 issues so far for £10 and, with volume four issue three being announced just after the show, that means show visitors who purchased the compilation got to see the latest issue before anyone else. Also available from Christopher was the long-awaited **55 BBC Micro Books** CD, also priced at £10: a collection of digitally remastered programming books as PDF and HTML.

Taking up what seems to be its customary position in the corner of Cedar Court's Cypress Suite, Risc os Open Ltd was next, where Ben Avison, Theo Markettos and Rob Sprowson were demonstrating and discussing the latest versions of Risc os 5.

[The pre-show buzz said Ro 5.2 would be



Martin Würthner, unusually, had no new versions of Artworks or Easywriter for this show – presumably because he was still in new-dad mode after the arrival of his firstborn, Anna Sophia, last Sept. 26.

released at Wakefield, but it came out only in August. —Ed.]

Netsurf 3 born at the show

In the opposite corner, the Netsurf developers launched version 3 of the browser – a version so new that they actually finished it off at the show. One of the biggest changes in **Netsurf 3** is that it now uses a new Document Object Model library, Libdom, an implementation of the W3C Dom, which is the foundation on which a fully dynamic layout engine can be built. This means the content, structure and style of web pages can be accessed and updated, for example by Javascript within the page itself, with the results being incorporated back into the page, and so displayed to the user –

an important step toward the long-awaited Javascript implementation in Netsurf.

Nearby, Steve Fryatt demonstrated **Cashbook**, **PrintPDF** and all his other software – notably including **Locate 2**, a flexible file-searching app that we had previously seen only as a test version.

Vince Hudd vince@softrock.co.uk



Vince Hudd, author of Webchange and other Soft Rock Software, lives in Bristol, which is not far from this mag's HQ. When your editor had to abandon plans to be at Wakefield, Vince kindly agreed to act as Archive's proxy at the show. It was the first Wakefield that Archive ever missed.

Mystery about jpeg vs sprite size

Kevin's photo of the Cedar Court Cypress Suite on page 11 was printed from a Jpeg wrapped in an Artworks file – which is Archive's normal way of handling photos in Impression. As an experiment I converted the picture to a sprite (using David Pilling's !Scan application). The sprite file is 1130K, compared to only 36K for the Jpeg!



Both files are on the Archive website – this calls for some technical discussion by Archive members. —Ed.

Another slice of Pi

*John Schild brings his
"First Steps into Pi-Land"
(Archive 23:6) up to date, with
a few digressions on the way.*



John Schild

It happened at last: the Iyonix died. In spite of a new power supply, new graphics card and a harddisc about which !DiscKnight could only report "disc is good" after some weeks of hesitant starting, one day it absolutely refused to boot and so demoted itself from crouching Panther to dead dog.

That meant, of course, that its internal Harddisc 5 backup offered no assistance; but with help from !Lanman98 (version 2) I had regularly backed up to networked storage, so my trusted apps and docs were safe(ish). A while after I wrote my "First Steps" article (*Archive 23:6*) my Raspberry Pi had been relegated to the toy box but now it was time to bring it out and test its readiness to take over as Risc os workhorse.

Lament of a lagger-behind

I was immediately, and deservedly, punished for not regularly checking and upgrading my Pi installation. Some of the limitations of the Pi as a serious work machine which were revealed in my earlier explorations now became serious.

Printing had always appeared to me to be one of Risc os's black arts, best left alone when it was working – which wasn't very often now. It had not much troubled me that copying, moving and saving files habitually broke down and stiffed the little box; now it was dire! And just for good measure, while the Iyo was serving out its

notice and the Pi was resting I changed ISP to benefit from Cornwall's new access to FTTC fibre broadband, so all my fixed IP addresses were now in the wrong range.

The SD card on which I had carried out my earlier tests was still in place with the official OS beta release of March 2013, no longer "clean" but subjected to a number of unrecorded, probably clumsy, updates. I had some writing I had promised to complete to a deadline, fortunately saved on the NAS and reachable once I had amended the server address.

But within an hour I was in total panic. I couldn't print, most of my attempts at moving and copying files failed, freezing the OS and losing any unsaved work. I had been mistaken: the DIY Pi is a toy, it would never be a reliable tool. My overheated brain stalled, unable to decide which way to turn to purchase a supported machine in a box: R-Comp, CJE or Virtual Acorn?

Eventually calm returned, the phone call was deferred, I transferred my unfinished script to Ovation on the PC, met the deadline, and in a contemplative fashion, began to work out my options, which basically meant starting again from scratch.

In a tranquil frame of mind

So this is what I did:

I dug out my second Pi (a Mark 2 board with a 512M rom) and promoted it to "lead" Pi. Using the Windows XP machine, I found the latest official Risc os RC8 download from the downloads page at the Raspberry Pi website. With the invaluable Win32 Diskimager, I transferred

the rom image to a new Transcend 4G SD card and offered it up to the new Raspberry Pi. It booted (as it should!) and, following the routine I described in the previous article, I got it running and onto the network. But having assured myself that it would run I decided to take another, more risky step.

Turning up the risk

I turned now to the Risc os Open website software page and found the then current (June 27) alpha development Pi rom image. I was intrigued to note that the OS version number had moved up from 5.19 to 5.21, but thought no more about it. I downloaded and opened the zip file with !SparkFS and parked it in the Risc os Pi root directory.

In the same directory I Shift-doubleclicked !Boot to open it. From there I opened the Loader directory, renamed the resident RISCOS/IMG file to something else; renamed the parked Risc os file as RISCOS/IMG (the capitals matter, I understand), dropped it into the Loader directory, closed all apps and directories and re-booted the Pi. It failed, complaining that !Boot was experiencing trouble but generously offering me a * command. I used this to key in *desktop, and was unsurprised but marginally reassured to be confronted with the fuzzy lo-resolution but still functioning desktop that I had seen before.

The other boot kicks in

That was when the penny started to drop: somewhere I had read that the upgrade of the rom image from 5.19 to 5.21 required an upgrade of the Harddisc4 image and this I had neglected to do. In Windows XP I downloaded and unzipped the latest stable Harddisc4 image, and using !Lanman98 transferred it to the Pi.

Driving almost blind I opened Harddisc4, and found the !Boot file. I set the file transfer options to include "newer" and dropped the new !Boot over the old one and re-booted.

Et voilà: I now had a good-looking high-res 1920×1080 desktop, and the Pi is up and running, ready to gallop, with the most recent software available.

I have since learned that I was luckier than I deserved. The Rool advice was, and still is: *install the new boot sequence first and gradually add back components from the old !Boot, testing as you go.* I was warned that the method I used might well have left abandoned bits of the old !Boot lurking there waiting to bite.

But, serendipity prevailing, it is good to be able to report that from this point on, the Pi running Risc os 5.21 has been stable. Saving and loading, moving and copying files, has proved secure and reliable. In short, I now have a working Raspberry Pi in everyday use. The loss of my once-reliable Iyonix workhorse now seems rather more trivial!

Another successful stride

There is one further step forward to report. In mid-July the Raspberry Pi download site released a new RC11 Risc os Pi image, complete with the new !Boot software. There was no further need to stick with my frigged !Boot upgrade, so I downloaded the image file, loaded it onto a new Sandisc class 10 SD card (for greater speed) and offered it up to the Pi. Following the recommended route, I began to put back, as needed, boot choices and tasks from the old card. I am writing this at the beginning of September, having on August 19 upgraded to the latest beta Pi rom – and I have no ill effects to report.

Another slice of Raspberry Pi

I may have given the impression above that I met with no headaches or hindrances on the way to a fully trustworthy Pi, but that would be an illusion. I conclude with some notes about the stickier bits of the journey in the hope that they might be reassuring to other DIY Pi-oneers.

Printing woes

Printing problems came “not single spies but in battalions”.

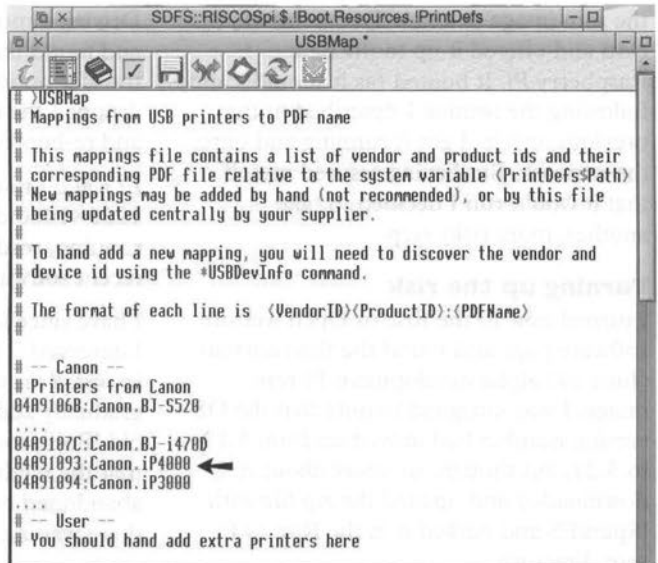
I have three printers, an elderly Canon iP4000 which I use for cheap and cheerful small documents, an Epson Stylus Photo R300 which I use more or less exclusively for high-quality photos, and the unwieldy bulk-carrier of a Samsung 550N network colour laser printer. When the Iyonix died and I urgently needed a printer none of them would play ball. What follows is an account of my experiences digging myself out of trouble.

What's this USBmap then?

The Canon printer presented an interesting challenge and led to a new discovery. I had never looked at the file `!Boot.Resources.!Printdefs.USBmap`

As delivered with a new Pi install, when `!Printers` is loaded, if a printer is switched on, attached and recognized – by its unique USB vendor and product ID – by its entry in the the `USBmap` – its printer definition file will be found and the printer will be run.

I found this out the hard way. The iP4000 would print, but halfway off the top of the



```
SDFS::RISCOSpi.$!Boot.Resources.!PrintDefs
USBMap
# Mappings from USB printers to PDF name
#
# This mappings file contains a list of vendor and product ids and their
# corresponding PDF file relative to the system variable <PrintDefs$Path>
# New mappings may be added by hand (not recommended), or by this file
# being updated centrally by your supplier.
#
# To hand add a new mapping, you will need to discover the vendor and
# device id using the #USBDevInfo command.
#
# The format of each line is <VendorID><ProductID>::<PDFName>
#
# -- Canon --
# Printers from Canon
04A9106B:Canon.BJ-S520
****
04A9107C:Canon.BJ-i4700
04A91093:Canon.iP4000 ←
04A91094:Canon.iP3000
****
# -- User --
# You should hand add extra printers here
```

USBmap: If a connected printer is identified here, its printer definition file [PDF in the old Acorn sense of the abbreviation] will be selected.

page. After hours of painful head-scratching I eventually discovered that the iP4000 printer definition file I had copied across from the Iyonix and installed conventionally into `!Printers` (via the menu item `Install new printer`) was being ignored, while the printer definition lodged under `!Printdefs.Printers.Canon` and listed in the `USBmap` was preferred, but turned out to be an older version. Replacing the older with the newer version did the trick – I could now print!

Printspool to the rescue

I have made another interesting discovery in trying, at first unsuccessfully, to print from the big beast of the Samsung network printer. On the Iyonix I had successfully used `RemoteprinterFS` (a module by Stefan Bellon, sbellon.de), but had failed to get it operational on the Pi. The interesting breakthrough was the discovery that I did not need to go that route.

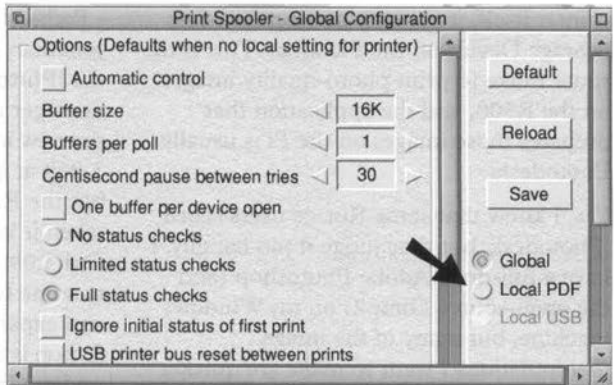
I've long been a fan of Ian Hamilton's !Printspool (hamiltoni.pwp.blueyonder.co.uk; current version is 2.18d). What I had not previously understood is that it is a viable alternative to RemoteprinterFS.

Load !Printspool; from the menu choose Set configuration. In the right-hand list, select the radio button Local PDF. *[Here the abbreviation means printer definition file, of course. To avoid confusion, Archive will abbreviate this meaning as "printdef". —Ed]* In the dialogue which now opens, enter the fixed IP address of the printer (fixed from the printer's set-up menu – in my case 192.168.1.80). Further down the dialogue select LPR interface. *[You can also tick automatic Load and quit with !Printers.]* Click Set and Save, and then as long as Printspool is running, the network printer will obey.

It really is as simple and reliable as that. Thank you, Ian Hamilton!

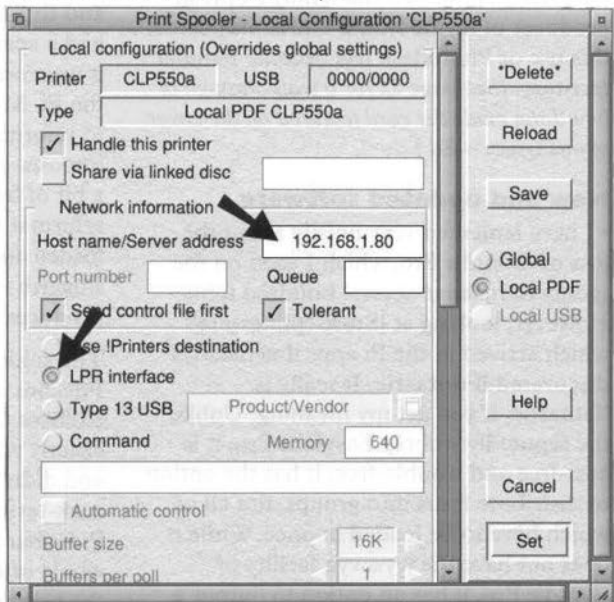
Photodesk ain't broke

So far I haven't been able to print via Gutenprint using the Epson Stylus Photo R300. It goes through the motions and chunters away happily but it produces only invisible ink, and I still don't know why. But at least for the moment I am not too bothered. This Epson printer does what it name suggests: it prints photographs really well, and this is almost its only duty when connected to my Raspberry Pi. Because I have never



Above: In Print Spooler's Global Config dialogue, select Local PDF to start connecting a network printer. *[Again, it's PDF in the sense of "printer definition file".]*

Below: Then, in the Local Config panel, enter the printer's fixed network address and select LPR interface.



succeeded in getting photo prints of the required quality out of Risc os !Printers, I now take this different route.

When the R300 printer is connected to the Pi through its USB bus and a memory card is inserted in the available slot in the

printer itself, it pops up an icon for Mass Storage Device on the Piconbar. This is the route I take to print photo-quality images on the R300, and the application that prepares those images on the Pi is usually Photodesk.

Yes, I know that some Risc os users scorn !Photodesk, but they judge it too harshly. I have a full-dress Adobe Photoshop (and the open-source Gimp2) on my Windows machine, but many of the image manipulations I want to make are quicker and easier on Photodesk, with no impoverishment of image quality. Really!

For example, because the R300 prints right up to the edge of the paper and I have been appointed the family's official producer of birthday and Christmas cards, this use of Photodesk has become familiar territory. *[See page 22 for a walk-through of how John made the card featured on the cover of this issue. —Ed.]*

New and updated software

- I have lamented very noisily about the loss of !Fontdir Pro, which I used on the Iyonix to manage several hundred fonts. However, looking at !Store (plingstore) which arrived in the Pi apps download, I discovered !Fontastic. It really is !fantastic, if you get my meaning. Unlike the reputedly wobbly Easyfont Pro, it is cost-free and trouble-free; it has the option to distribute fonts into groups, not all of which have to be loaded at once. While it does not have the wysiwyg facility of Fontdir Pro, it has an option to output a drawfile or an Ovation Pro DDL file to make an easy-reference printout of all my fonts. I like it! And I have stopped weeping for Fontdir Pro. Fontastic is by Richard Windley; download the latest version (0.14 dated 2012) from reallysmall.co.uk

- Perhaps the best bit of Pi news is that Jonathan Duddington has open-sourced the !Pluto email and news reader and manager as well as recompiling it to run on the new architecture. You can find version 3.06b at sourceforge.net/projects/plutoemail, but the Risc os community has already taken it in hand and produced a 3.08 and 3.09, and I think we can look forward to some further ironing-out of its wrinkles and expanding of its capabilities. Martin Avison is trying to prevent private-enterprise updates from fragmenting by setting up a web page at www.avisoft.f9.co.uk/Pluto.htm – certainly worth keeping an eye on.

- !Hform, the Risc os drive-formatter has always scared me. What if I get it wrong and format the wrong drive? However, I had a spare 230G Seagate portable drive; it was formatted as NTFS and useless to the Pi. My Windows XP machine refused to reformat it to Fat32 because the OS imposes a size limit. But I was able to load a bit of free third-party software to reformat the drive to Fat32 under XP. It loaded and ran on the Pi but was incredibly sluggish – because, as I understand it, it produces a huge file-index map.

The solution seemed to be to format it as Filecore using !Hform (buried in the \$.Utilities.Caution directory), which did exactly what it promised. It took a while and didn't seem to know when to stop soak-testing, but it now loads to the Piconbar as SCSI:4. I have copied the whole of my ex-Iyo harddrive to it from the network-attached storage, and to date it has been absolutely safe, secure and reassuringly fast.

Where next?

The experience of some months running a Raspberry Pi as my Risc os workhorse – principally for document-production,

email and simple (Netsurf) web-browsing – has shown up its main limitation for my work: the 2G size limit of the Pi-boot drive.

It pleases me to keep my always-in-use applications and utilities on the boot drive, with the consequence that work files, research notes and internet downloads towards a writing project (like this one!) have to be run from a second, larger, drive. This always runs the risk of a breakdown during file moving and saving. I know that CJE Micros will sell me a fast hand-crafted 32G SD Pi-boot card with Risc os already loaded, and I am held back from that course of action only by my wish to stay and play, for the moment anyway, in the DIY sandpi(t).

I am also aware that the Windows XP computer remains a vital hub of my Risc os enterprise, indispensable for creating Pi-boot SD cards to run on the Pi. I would like to bring this operation in-house, reducing or removing dependence on Windows, and I believe that help might eventually be at hand.

I received a copy of **!Clonedisc** from Piccolo Systems (Ben Avison) as part of the NutPi promotion, and tried to use version 1.00 to copy disc images to SD cards – unfortunately so far without success. Already Clonedisc 1.01 has been issued, and it is anticipated that a new incarnation might soon come forth, allowing for partition management – which would, for example, allow the whole of a 16G or 32G SD card to be used by DIY Risc os Pi-oneers.

Still a two-gig ceiling

Meanwhile, the 2G limit remains, and I have disciplined myself to making frequent backups of work-in-progress. The good

news is that keeping a good-quality SD card plugged into a card reader in the Pi's second USB slot, and using that as my working drive, I have experienced zero filer hiccups. This is also true of the 230G Filecore-formatted portable drive I plug when needed into the 10-way Trust USB hub which connects the Pi to its keyboard, mouse and peripherals.

The only disappointment is a 32G Fat32-formatted USB pen drive that I had intended as my working drive. It has not proved entirely reliable. It occasionally falls over when transferring a large file and crashes the OS in its descent.

Where next? While I wait for a 32G boot drive, I think maybe another project beckons: finding a Sata solid-state drive and a Pi-happy Sata-to-USB adaptor. Has anybody out there got one working? Please let me know.

£1,000 down to £25

Overall this has been an interesting, sometimes even mystical, journey. The last word has to be one of praise – not only for the RoOL corps of high-geek-power amateurs and professionals but also for the prophet who foresaw that in 2013, a diminutive pocket-money computer might step into the shoes of an Iyonix that cost £1,000 a decade ago and fill them: *Wow!*

John Schild john@schildren.co.uk



Now retired from pounding the pulpit steps, the Revd John these days lives in Cornwall and keeps his Pi busy at creating study materials for church groups and Readers-in-training.

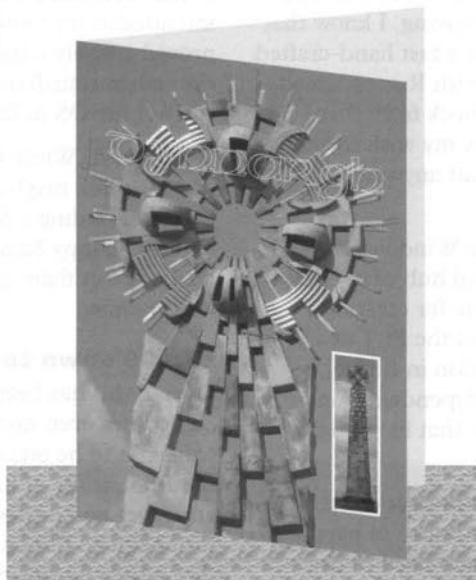
Greetings from Photodesk

John Schild

I have an Epson Stylus Photo R300 printer and by undeserved good luck discovered that it can take the Jpeg output from a piece of image software and input and print it as though it had come straight from the camera's memory card. This has three merits:



- it eliminates some of the colour-accuracy problems of conventional Risc os printing
- it allows photo images to be processed in a variety of ways while preserving the photo quality that may be lost in !Printers
- it will print right up to the edges of A4 glossy photo card – a real bonus for printing professional-looking cards.



Nut Pi nuggets

The project I want to describe was all carried out on a Raspberry Pi computer, using only !Photodesk 3.12 from

the Nut Pi collection of software and good old !Draw, which is part of the Risc os rom. To keep reader fatigue at bay I have had to assume some knowledge of the basic workings of Photodesk. All images were saved as Jpegs.

Preparing the photographs

In 1859 Isambard Kingdom Brunel opened his Royal Albert Bridge across the Tamar. Never slow off the mark, Cornwall decided 2013 was probably about time to say *Dynnargh* – Welcome to Cornwall – by erecting at the Saltash end of the road and rail bridges a gigantic, high-tech Celtic

Cross. Never one to miss a free photoshoot, I trotted along to look for the killer photo. I found two: a wide (or do I mean high?) shot emphasizing the height of this new apparition and a 20x zoom shot of the business end.

I decided to do my bit for Cornish tourism by building these two images into a welcome card – easily converted for another occasion such as a birthday. Although I do have the big Adobe beast Photoshop on my Windows computers, Ro's humbler cousin Photodesk has much to offer. And

candidly, for some jobs I actually prefer Photodesk: this was one of them.

I transferred the two images to the Pi, opened up Photodesk and offered the two pictures up to its clipboard. Aware that the final print will be only A5, I began by

An impression of the finished card, generated by !Photodesk's "distort" function and colouring the back with the "filled parallelograms" drawing tool.

I did this, rather than photographing the card on the mantelpiece, to demonstrate that Photodesk can distort images and paint with textures. I can't really say it is pretty, but the technique worked!

scaling both large photos by half without losing significant definition, rotated them and re-saved them under new meaningful filenames. Each stage was similarly saved to disc so that the inevitable mistakes would not invariably drive me back to square one.

Crop to 'A' ratio

I'll explain why later, but the next step was to take the major image – the business end of the cross – and using the cut-and-copy facilities of Photodesk, I cropped the image so that the X:Y pixel ratio was 1:1.414 (portrait). Because I was blessed with good light and no camera shake, this image needed no more work so I saved it and set it aside. *[The figure 1.414 of course is the square root of 2, which the clever ISO format uses to retain the same shape when A4 paper is folded in half to make A5 and when A5 is folded in half to make A6 (etc).—Ed.]*

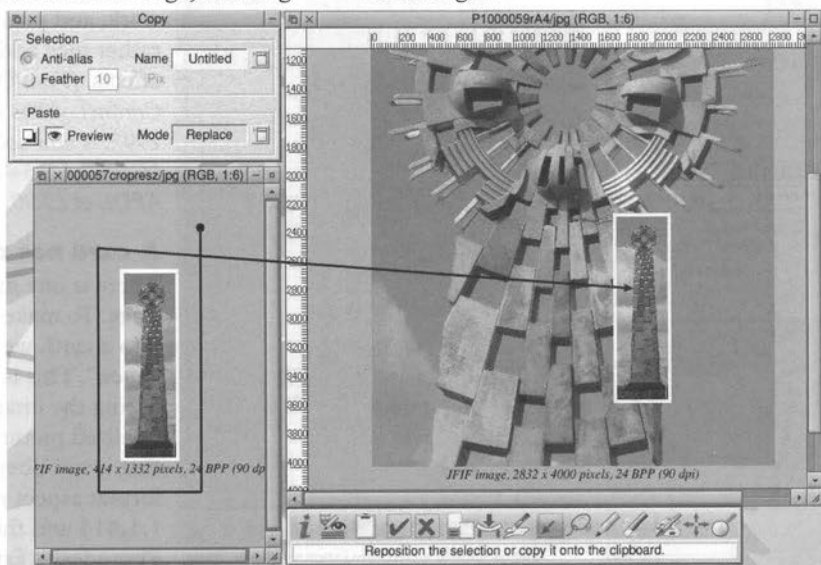
The next stage was to create an inset in a quiet corner of the main image, showing the whole cross but without the distractions of passing traffic and the odd stray dog. This photo I now opened on the “easel” or work area and cropped it severely to just outside the height and width of the cross. It was still too big to copy

across as an inset, so I further reduced its size using the Image > Resize option from the Photodesk menu until experience suggested it might be about right.

Give the inset a border

To look “intended”, this little inset needed a white border, so, using Photodesk's Extend option (located rather improbably under Crop, but with hindsight it makes inverse sense), I added a white border of about 50 pixels all round. It was now ready to be copied across to the main image and placed where I wanted it.

I opened each image in a separate window, selected the main image, invited Photodesk to copy to this window, dragged a box right round the image to be inset, checked that the Paste section of the Copy dialogue was set to Preview and Replace mode, then picked up and dropped this image where I wanted it over the main image. Using the “hand” pointer I kept moving it until it looked right.



Draw a box round the area to be copied and drag it to the main image. In the Copy dialogue, make sure that Preview and Replace are active.

Once I was sure I had got the scaling and position OK I Select-clicked the “tick” box, pressed the spacebar to confirm, Photodesk peeped its characteristic peep, and I had my composite picture. Had it looked wrong, I could easily go back and make some adjustments to the inset picture and keep trying till it satisfied me.

It was quicker and easier than the write-up suggests.

Now for the welcome

A nice feature of Photodesk is that vector images from Draw can be dragged over a bitmap image, scaled, angled and “rendered” into the photo image.

I took the Cornish word for welcome, *Dynnargh* (pronounced more like “diner” than “dinner”), and found an appropriate Celtic-style font. I typed *Dynnargh* as text into Draw, selected the appropriate font and sensible size, chose Convert to paths and

Group, and set Line width to 1 point.

Back on Photodesk I used the colour-picker syringe to identify the RGB values of one of the gold areas of the Cross close-up and copied these values to Draw. I then set the Fill colour of the letters to “none” and saved the drawfile. From the filer I dragged this onto the composite image in Photodesk, where it settled along the bottom at maximum scale. Using the graphic-rendering dialogue I scaled it by eye and dragged it to where I thought it looked right.

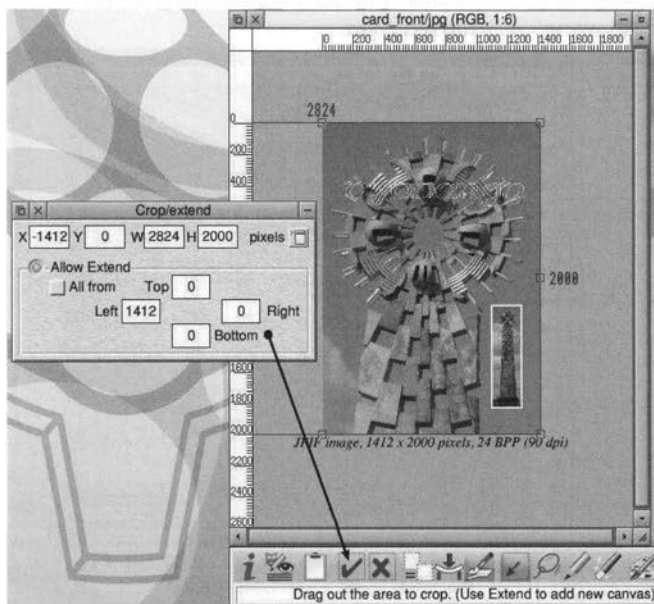
Once I was satisfied, I invited Photodesk to “render” it, and that completed the image-building: I now had the front of my card.

Different outline fonts might require different line widths, line and fill colours in Draw to render to a convincing image; just experiment till you are satisfied. If you have a copy of the venerable **!FontFX**

application, this takes much of the effort out of the Draw work, and provides some rather nice additional effects. [*FontFX*, by David Coronel of The Data Store, 1997, is now sold by CJE Micros at £25+Vat or by APDL at £9.90. —Ed.]

A card has two sides

There is one more easy stage. To make this image into a card, we need a “back”. This is achieved by noting the image size of the finalized picture, checking that it still obeys the ISO-format aspect ratio of 1:1.414 and then, using Photodesk’s Extend function once more, add an exactly



To make the card’s back, enter the width by which the image is to be extended, then tick to accept.

equal and symmetrical area of white canvas to the left side only. We now have a printer-ready photo greeting card.

If you are serious about greeting-card production there is nothing to stop you creating your own production logo in Draw format and rendering this to the back of the card.

Now let's print

The card needed to have "A"-format dimensions because it is not going to be printed by Risc os; instead it will be transferred as a Jpeg directly into the Epson R300 printer.

Here's a nice feature of this printer when it is switched on and connected by USB to the Pi: if a memory card is plugged directly into the printer, it will show up as a drive on the Pi's iconbar.

Open the root directory of this drive and copy the photo image into it. Make sure its filename has the .jpg extension. It pays to remove and replace the memory card in the printer slot at this point; otherwise the printer ties itself in knots.

The printer will display information about this image and ask how it should treat it. Instruct it to print one copy, A4, glossy, no border, and off it will romp.

The really nice thing is that although there might be some slight cropping of the edges (so keep important detail just a bit inboard) the fold-line of the card will fall exactly right, and you have that professional-looking greeting card all ready to go.

A6 cards can be produced by creating an identical area of white space below, cloning a copy of the front image to it and proceeding as above; you get two identical cards out of one A4 sheet – ideal for producing Christmas cards!



Photodesk footnote

I have sometimes found that the printer sulks rather than obediently printing; this has usually been

because the image was just too big, and had to be reduced a bit – keeping the aspect ratio constant.

I suspect that one reason why some would-be users give up on Photodesk is that they have not found the best settings for their computer. It is fussy about dynamic areas and image cache, and doesn't like !Twain being auto-loaded. I could probably have chosen some better values, but I have found that setting Photodesk\$DynamicAreaSize to 512 in the !Run file, and image cache to 100022K in Preferences works well for me on the Pi, although these values were not ideal on the Iyonix.

While my Epson R300 persists in printing only invisible images from RISC OS, this partnership with Photodesk keeps it in regular use so its ink cartridges do not dry out. It is the only printer on which I use manufacturers' own ink cartridges – all six of them.

Photodesk 3.12 is available from CJE Micros at £84; upgrades from previous versions sell at proportional prices, and there is also a free demo version.

A full version of Photodesk tailored to run only on the Raspberry Pi is included among the 20 commercial applications in the excellent Nut Pi collection, all for £35 from Risc os Open Ltd.

John Schild john@schildren.co.uk

Coming back to Ro

This article began with an email to the contact address on the Archive website. The writer identified himself by saying: "I'm a 'revert' – back after 20 years thanks to the Raspberry Pi. There was a mention of two free issues ..."



Malcolm Hussain-Gambles

My introduction to Risc os came in the early days of the Acorn Archimedes, when my mother brought home an A310 from school over the summer holidays. It ran not Risc os but its predecessor, Arthur.

I was already using my mum's BBC B, programming in Basic and doing a little in 6502 assembler. After a couple of years arrived the A3000 which my parents bought for me. My parents used Impression Junior to do some desktop publishing.

My love for Risc os continued. I remember upgrading to an Arm3 processor, 4M of ram and an 120M harddisc over the years. I dabbled in some fairly complex programming: a spreadsheet program, 3D vector graphics, a compiler and obviously the usual games. I have fond memories of *E-Type*, *Elite*, *Holed Out*, *Chocks Away*, *Interceptor II* and my favourite, *Stunt Racer 2000*.

After the A3000 I eagerly awaited the next generation of Risc os machines. The RiscPC came out, but it was way too expensive for me as a teenager, so I hung onto my A3000.

I eventually landed my first IT job, and Acorn was about to release the Phoebe. I was excited, as I finally could save enough money to buy a new computer – then came Acorn's Black Thursday in 1998, and

all my hopes were dashed. My heart sank, and I left the Risc os world for what I assumed would be forever.

A new chapter

Moving forward a decade and a half to the present, I bought a Raspberry Pi – for the money you can't go wrong.

I installed Debian Linux on it (as I am a Unix-Linux guy). It worked well enough, but I was heavily disapointed at the speed of X. Running Firefox and Thunderbird on it was painful beyond belief. I had a bit of a play, but put it on a shelf and went back to my quad-core 3GHz Intel processor with 12G ram and PCI SSD harddisc, which was much more bearable – again Linux.

A bit later I heard the news of the alpha release of Risc os for the Pi by Chris Hall. I felt like that 12-year-old kid getting his hands on my A3000 again!

I couldn't believe the speed difference between Risc os and Linux on the Pi. I had expected to be disappointed, and booted the Pi with a sick feeling in my stomach.

Blown over

After I had managed to get my jaw off the floor and stopped swearing in admiration (thank goodness my wife and kids weren't around), I was amazed at what I saw.

I also realized if I wanted to use Risc os again, I was going to need some decent applications. I noticed R-Comp was still going – I was amazed at this too. I had met Andrew Rawnsley way, way back when R-Comp first started, I think.

Once I had most of the applications I needed and the Rool compiler I started using the Pi as my primary desktop – I even

splashed out and bought Artworks and Easiwriter from MW Software. As a kid I had always drooled over Artworks, and as a slightly bigger kid I could afford it – lovely!

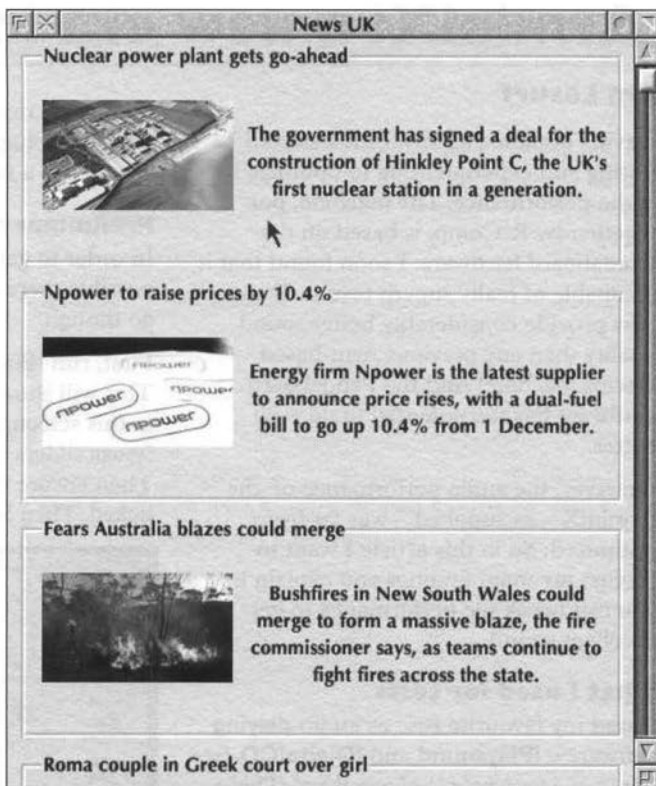
I soon realized that while the Pi was nice, it was a little too slow for me. I have since upgraded to an ArminiX – the cased Pandaboard from R-Comp.

Overall

My impressions of Risc os are generally amazing, although clearly there are deficiencies. The team at Rool have my deepest respect. I hope the Raspberry Pi will bring more people to Risc os so that the resources that this lovely OS desperately needs will bring it forward and fill in the gaps that are there.

I wish I had more time to spend coding and contributing to Risc os. I've started programming again on Ro, but my time is extremely limited, which is a shame.

I don't want a desktop OS that can support booting off a SAN fabric and have eight paths to each disc with support for everything under the sun. I want a lightweight, reliable and easy-to-use system – which is exactly what Risc os gives me. I really hope that this doesn't change as it goes forward.



Since writing this article, Malcolm has released **!NewsUK** and **!WeatherUK**, which display up-to-date information from the BBC and the Met Office. "They're not amazing apps, but kind of nice and help to fill the huge void," he says. They are available free from !Store (plingstore.org.uk). Supplied with them is a new version 1.40 of Acorn's !ChangeFSI, updated in August 2013.

Malcolm is now working on !Rabber, a Risc os Jabber client.

Calling Gavin Stark – "If only I could get in contact with Gavin Stark again," Malcolm says. "He was the ultimate badass Risc os coder. I've still not met a programmer who could hold a candle to him – and I've met a lot!"

Malcolm Amir Hussain-Gambles
malcolm@paymentlabs.com



Malcolm was Gambles when his story began. The Hussain bit is from his wife's side. They moved to Hull in December. This rather dire mugshot is a last-minute snap from his phone.

ArminiX: a new era in RISC audio

Jim Lesurf

Having obtained an ArminiX I started testing and experimenting to optimize its audio performance. The machine, put together by R-Comp, is based on the Pandaboard hardware. I soon found that it is capable of really superb results. It can now provide considerably better sound quality than any previous Arm-based RISC OS machine! And the Pandaboard hardware has the potential to do even better.

However, the audio performance of the ArminiX – as supplied – was far from optimized. So in this article I want to discuss my main findings and explain how you can tweak the performance to get excellent sound.

What I used for tests

I used my favourite RISC OS audio-playing software – **!Playsound** and **!DigitalCD** [see box at end for where to obtain them]. These both work well. But – as with the setup of the OS – they need to be used with care to get the best results. For my tests I used a series of Wave sound files that I had created with **!Wav_Gen**. I used these because I know they provide high-quality test material. (You can find a copy of **!Wav_Gen** on the software page of my www.audiomisc.co.uk website, along with various other items of audio software.)

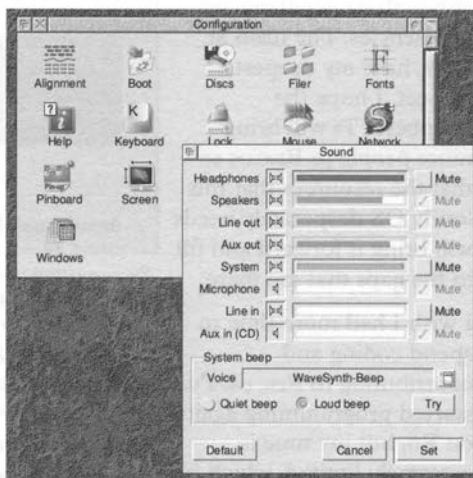
What I write in this article is based on the version of RISC OS 5.19 and its HAL (hardware abstraction layer – the code that glues the OS to the hardware) that was supplied to me and other users up to the time I wrote this article. This may well change in the future to provide improved behaviour. So some of the things I say in

this article may then become obsolete. When that happens I'll write about the changes in a future article.

Preliminary steps

In order to get things working as well as possible there are a set of steps you should go through.

First, run **!Boot** and select the Sound item. This will show you a window like the one in this screengrab. Drag the Headphones and System sliders all the way to the right. Then ensure that their Mute buttons are *not* ticked. Then Set these choices. You should



Configuration settings for sound.

then shut down and reboot for the changes to come into action; otherwise you may find that you can't get any output at all or the volume will be wrong for what follows.

Once you've done this there are some other changes I recommend.

Sample rate

First, the default system "sample rate" on

the ArminiX is 44,100 samples/sec. This is the rate used for Audio CDs and also for most of the other music files that are around, such as MP3. So it may seem the sensible choice. But – as things stand when I wrote this article – in practice that rate does not work correctly. To examine the sound settings I wrote a simple program that I called **!Soundcheck**. This interrogates the system and lists what it finds. I've provided a copy to go on the Archive website to accompany this article. I have also provided some of the other programs I'll mention using below.

 **Files:**
see p.2

When run, !Soundcheck opens a taskwindow to report its findings. An example is shown in the screenshot below. Among the values listed, two are most significant here: the audio system's current sample rate, and the "volume" setting. These both need to be changed to obtain optimum quality.

At present, when you want to play audio files on an ArminiX that have their data at a sample rate of 44,100 you should set the system rate to 88,200 samples/sec – in other words, to exactly twice the actual

rate of the file. You can do this with a simple Basic program I've provided. Then run !Soundcheck again. It should now tell you that the current rate is 88,200 rather than the default of 44,100.

Change 'volume' too

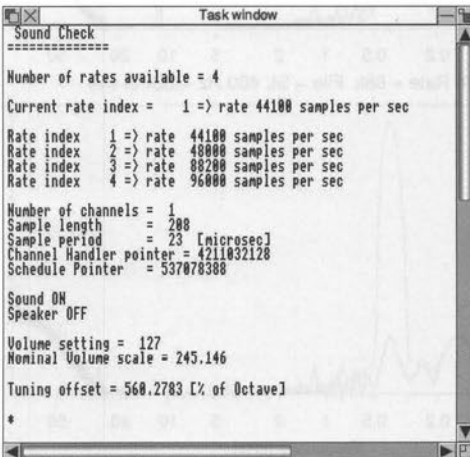
The second value that needs to be changed is the system's Volume value. By default this is 127, which has the effect of scaling actual sample values by 245 percent. This will mean that loud sounds in files (or streams from net radio) you want to play may be clipped and thus distorted. To cure this you can either open a taskwindow and type in the command `*volume 100` or set up an obeyfile to issue this command. This will reduce the gain to a value that avoids the risk of loud sounds being clipped.

Actually, a volume of "100" nominally gives a gain of 76 percent, not 100 percent (crazy, really!) and I can choose a `*volume` value of up to 103 without seeing any sign of clipping. But 100 seems a convenient round number to remember and use. So I have adopted it for now, and suggest others do the same unless they have a reason to set a different volume level for the system.

Alas, at present these changes in rate and volume will be forgotten when you shut down the machine. So you either have to repeat them or put some suitable files in the bootup of your ArminiX to apply them automatically each time the machine is rebooted. Among the downloadable files on the Archive website I've provided a zip of some programs I use.

Automate and beware

To automate these settings for the volume and sample rate, I put a copy of two programs – `Setvol100` and `Set88kHz` – into my directory `$.!Boot.Choices.Boot.Tasks`.



```

Task window
Sound Check
=====
Number of rates available = 4
Current rate index = 1 => rate 44100 samples per sec
Rate index 1 => rate 44100 samples per sec
Rate index 2 => rate 48000 samples per sec
Rate index 3 => rate 88200 samples per sec
Rate index 4 => rate 96000 samples per sec
Number of channels = 1
Sample length = 288
Sample period = 23 [microsec]
Channel Handler pointer = 4211032128
Schedule Pointer = 537078388
Sound ON
Speaker OFF
Volume setting = 127
Nominal Volume scale = 245.146
Tuning offset = 560.2783 [% of Octave]
*
  
```

The taskwindow that !Soundcheck opens.

Thus each time I restart my ArminiX it is set to use a sample rate of 88.2k and a volume setting of 100 (about 76 percent). This is a quick way to avoid some pitfalls that can otherwise degrade the sound. You can then use !Soundcheck to make sure the settings are as you wish.

Beware that some user programs may tamper with these values. So when you start using a particular sound application on your new machine you might wish to run !Soundcheck again to make sure all is well. In what follows I'll assume these changes to settings have been made. Note, though, that the need for these two tweaks might change in the future if the HAL or the OS is developed to deal with these details.

Out of the box

The ArminiX has two headphone-style jack sockets on its back. The lower socket is the output. With the system volume value set to 100 when I play a Wave file of a full-scale (0dBFS) sinewave I get an output of 1.03 Volts rms. (The upper socket is for inputs. This is connected but as yet there is no firmware to allow it to be used. For that, the ArminiX or Pandaboard needs the equivalent of the !AudioIn system used on the Iyonix.)

Figure 1 shows the results I can get when playing a test Wave file. This is a 400Hz left-channel-only sinewave tone in the form of a 16-bit 44.1k (i.e. Audio CD format) file. If I play this

with the ArminiX set to use its out-of-the-box default system rate of 44k I get two problems. One is that the output is mono, not stereo. What should only appear on the left channel appears on both.

The other problem is that there are some added high-frequency components. These are almost as loud as the intended 400Hz tone! This is due to an, erm, "feature" of the released version of the HAL. However, if I set the system rate to 88,200 both of these problems evaporate and the result is quite clean.

!Playsound adjustments

In fact !Playsound has two "preferences" that can help (I'll explain the reasons in another article). These are Stop other sounds

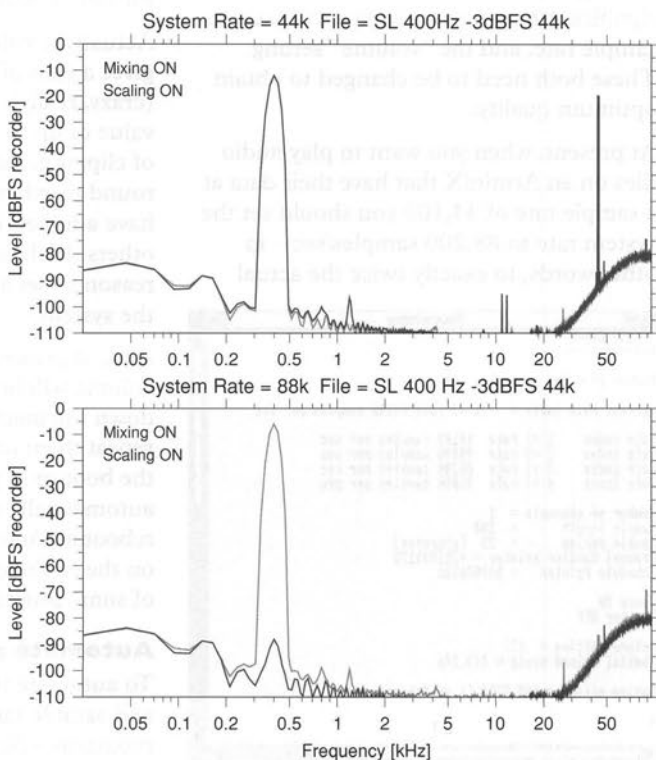
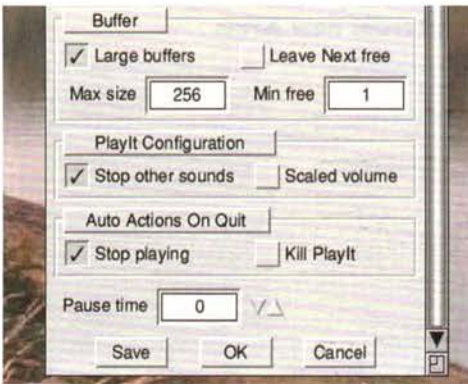


Figure 1 – A test Wave file played at 44.1k (above) and 88.2k (below).



Two of !Playsound's preferences will help.

and Scaled volume. Ideally, selecting "Stop other sounds" and not selecting "Scaled volume" can improve results. So I set these as shown in the above screengrab before doing some more tests.

Figure 2 shows the results when I played a couple of test files. Each contains a pair of sinusoids on just the left channel. Bear in mind that using a pair of high-frequency tones like this is a fairly demanding test of the kind used to try and catch out expensive and fancy hi-fi systems – this signal stresses the system far more than most normal music or speech.

The upper graphs shows the result when I played a Wave file that has a sample rate of 44.1k. The lower graphs shows the result when I played a file defining nominally exactly the same sine waves but recorded with

88.2k samples/sec. You can see that the 88k file removes the unwanted ultrasonic components at about 50kHz. In fact this 88k result compares well with many professional "hi-fi" DACs (digital-to-analogue converters) that I have used or tested in recent years. So the ArminiX hardware has the potential to work outstandingly well if used optimally!

The added ultrasonic components arise when playing the 44k file because the sound system and player has to "interpolate" the 44k data to play it with a system rate of 88k. The difficulty is that either the Risc OS audio-playing software or the OS itself tends to use plain linear interpolation. If the system rate is exactly

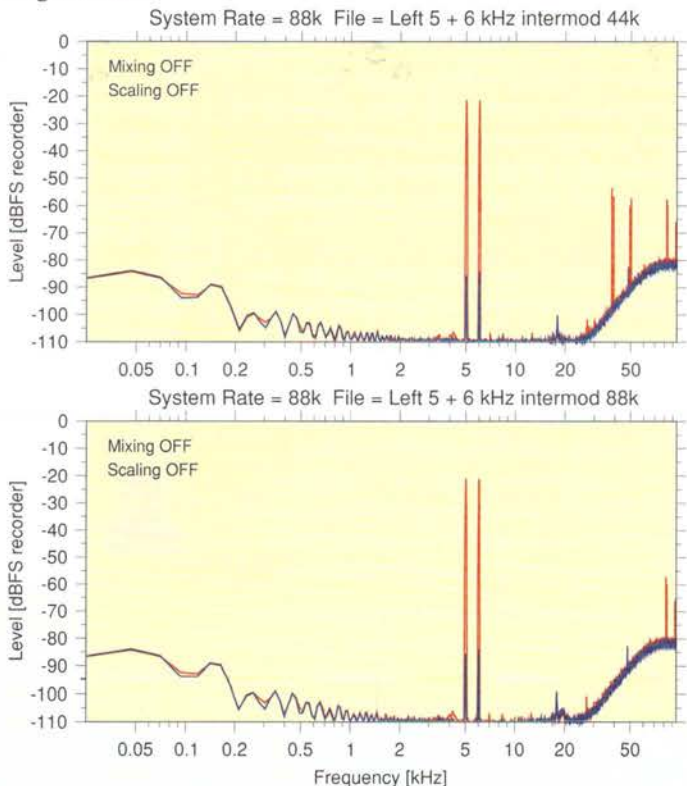


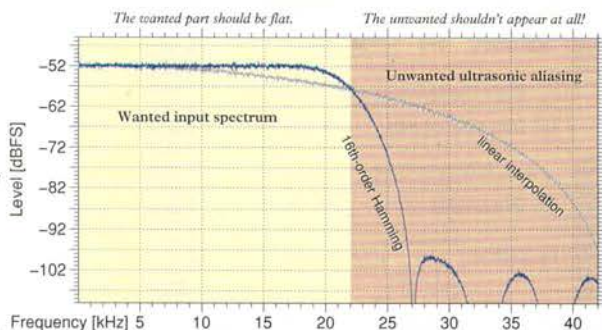
Figure 2 – After tweaking !Playsound, compare the results from the test files.

The state of play: Upsample for top sound from new Arms

Modern Risc os systems like the ArminiX and Panda Ro are based on new hardware. This offers the ability to play audio at high sample rates. However, at present most Ro software can't make best use of this. I have therefore released two new "demo" applications that use improved methods to "upsample" audio and get results that sound much better.

- **!DRplayer** is a very simple player for Wave files. You will hear them with better sound quality from machines like the Pandaboards that support system rates of 88.2 or 96k.
- **!Wav_upsampler** employs the same methods but generates a new Wave file at double the rate of an input file. You can then play that with the usual applications such as !Playsound and get better audio quality than before. The result should be the best audio quality ever provided by Ro hardware!

Further info and downloads: www.audiomisc.co.uk/software/ArminiX/Upsampling.html



double the file rate then simple linear interpolation works fairly well. Although it does generate unwanted distortion components, they appear at ultrasonic frequencies and tend to be moderately low in level.

Intriguingly, some audio fanatics say that this behaviour *improves* the sound quality! But others are rather doubtful and wonder if the ultrasound is even audible.

Best sound ever from Risc os

Technically speaking, there are much better ways to interpolate or "upsample", as hi-fi buffs term it. But, alas, Ro doesn't use them as yet, and they require much behind-the-scenes number-crunching.

Nevertheless, the results you can now obtain from a Pandaboard system when playing "Audio CD rate" files with the settings I suggest will be better than anything obtained from earlier generations of Ro hardware.

And, with luck, there will be further improvements in due course. I'll say more about that in my next article.

Jim Lesurf jcgl@audiomisc.co.uk



Jim's articles about "Sound on vision" were written to compare the sound quality of the Proms via Radio 3 FM against that from BBC4, which is a digital channel: www.audiomisc.co.uk/HFN/SoV/soundonvision.html

Downloadable sound applications



Mark Scholes put all his software back online in December, including **!Playsound** 1.74, at mark.scholes.org.uk/software/riscos. It requires Rick Hudson's !Playit, now maintained by Christopher Martin; a copy is on Mark's site.



André Timmermans' **!DigitalCD**, a general-purpose audio player, can handle MP3, Flac and Ogg Vorbis as well as audio CDs and internet radio. Steve Fryatt reviewed it in *Archive 18:11*. Version 3.06 is the latest: www.riscos-digitalcd.net



The ARM*ini*X

What does the X mean for you?

- ARM Cortex A9 at 1.2 GHz
- 1 gigabyte ram
- 32GB high-speed flash storage
- DVD read-write drive
- disc operations 4× Armini
- RISC OS 5.2?

Software bundle includes:

- Aemulor Pro
- !MusicCD



Interactive
RComp

www.armini.co.uk

phone 01925 75 5043

Illusions – and Jpeg compression

Gerald Fitton

Joni Mitchell describes herself as “primarily a painter, secondarily a musician”. It was she who coined the words for the song *Woodstock*:

*We are stardust
Billion-year-old carbon
We are golden
Caught in the devil's bargain
And we've got to get ourselves
Back to the garden.*

Of course you will know that we are all made of the dust strewn in space when a star or two exploded billions of years ago. Indeed we are made from “billion-year-old carbon”. Also, you will appreciate that the “garden” is the Garden of Eden; this is an allusion to changing our lifestyle to one that welcomes simpler, more virtuous, fundamental, even illusory, values. I would suggest that Joni, like so many singer-songwriters, is also a poet.

The Discrete Fourier Transform

In the last couple of issues of Archive I described how to use a Pipedream spreadsheet to find the frequency spectrum of a time-dependent, periodic wave. The Discrete Fourier Transform (DFT) finds the amplitude and phase of a set of harmonically related,



*It's life's illusions I recall;
I really don't know life at all.*

—from “Both Sides Now”

a hit in 1967 for
Judy Collins (1939–)

music and lyrics by
Joni Mitchell (1943–)

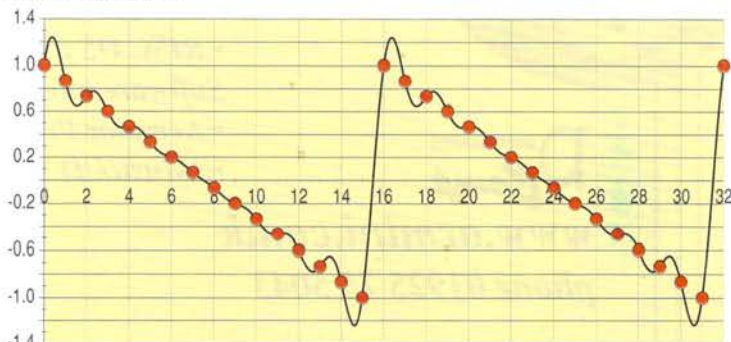
sinusoidal frequencies that can be used to reconstruct a wave passing through the original sampled points.

The first graph shows two cycles of such a reconstructed wave. You will see that the curve which passes through the original 16 points, 0 to 15, is repeated exactly for the 16 points numbered 16 to 31. The graph is repeated exactly every cycle.

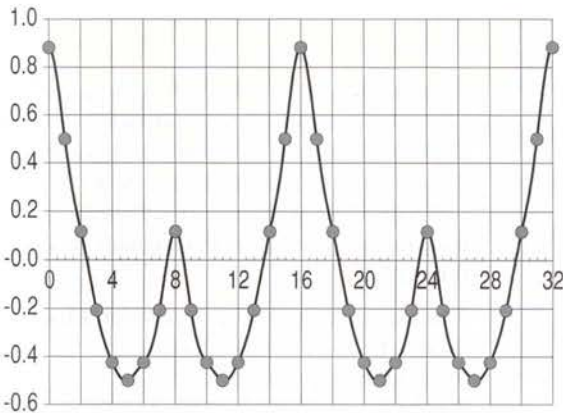
The Discrete Cosine Transform

Contrast the graph of the 16-point Fourier Transform with the graph of the 9-point Discrete Cosine Transform (DCT) shown in graph 2 on the next page. The original 9 points are those numbered 0 to 8.

Notice that, outside these original 9 points



(1) A 16-point Discrete Fourier Transform reconstruction of a periodic wave.



(2) A nine-point Discrete Cosine Transform reconstruction of a periodic wave.

(numbered 0 to 8), the graph for the points 8 to 16 does not repeat in an identical way. The graph for the points 8 to 16 are a mirror image reflected in the vertical line $x = 8$. Similarly, for points 16 to 32, the graph is a mirror image of the graph for the points 0 to 16 reflected in the line $x = 16$.

Interpolation and extrapolation

For interpolation, both the DFT and the DCT return an identical reconstruction. It is only when the sample is used to extrapolate outside the sampled region that these differences appear. The reconstruction from the frequencies of the DFT repeats exactly the same waveform every cycle; the frequencies of the DCT return a series of mirror images of the original waveform.

Why choose DCT rather than DFT?

Why am I bothering to discuss DCT at all when a DFT provides us with both amplitude and phase information in full? The answer is that the DCT is a much more efficient transform than the DFT, when extrapolation outside the original points is not required. The DFT requires

both cosine and sine functions whereas the DCT uses only the even cosine functions. Examples for which extrapolation is not required are the Jpeg compression of a digitally encoded photograph and the MP3 compression of audio.

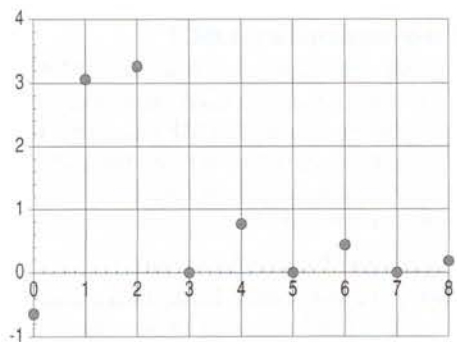
Compression is used extensively to reduce file sizes by discarding some of the fine detail which our ears, eyes and brains replace with what we expect to be there. The DCT carries out the data-processing necessary to enact this illusion with much greater efficiency than does the DFT because, unlike the DFT, all the sine components are discarded.

The amplitude spectrum

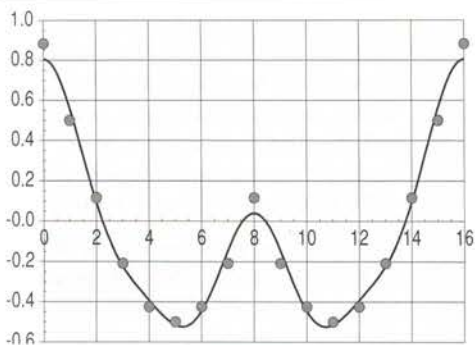
The amplitude spectrum of the 9-point DCT is shown in graph 3. The amplitude of the lower frequencies is greater than the amplitude of the higher frequencies. This is typical of Discrete Cosine Transforms, including multi-dimensional DCTs, but is more unusual for a DFT. This fact is what makes the DCT a more efficient means of compression than the DFT.

Reconstruction

In the reconstruction (graph 4), the original 9 points are those numbered 0 to



(3) The amplitude spectrum of a nine-point DCT.



(4) A reconstruction discarding four of the nine DCT frequencies.

8. In use, the original is reconstructed only between the points 0 to 8; I have included points 9 to 16 solely for reference. In this reconstruction, rather than use all 9 frequencies, I have discarded the higher frequencies. The resulting waveform is a close approximation to the original. The drawfile shows in red the original 9 points (plus the reflected points); the black line shows the reconstruction from only 5 of the 9 original amplitude components.

The error between the original and the reconstructed values is about 10%. As we shall see in the pictures below, this 10% error is not easily discernible to the ear nor the eye because we “fill in the gaps” with illusory, smooth lines. It is the illusion which we hear or see.

Two-dimensional DCT

I’ll stop discussing one-dimensional DCTs even though they are used extensively for creating the ubiquitous MP3 files that now fill countless portable devices. Instead I shall consider what is usually called a two-dimensional DCT.

My 2D DCT example, Picture 5, is a photo of a lady called Lena. This picture – usually called Lenna – is famous because it has been, and still is, used extensively to

test algorithms for processing images. [For the curious history, see page 5.]

Graphic 6, which looks a little like a galaxy of stars, is a pictorial representation of the Discrete Cosine Transform of this Lenna picture. The brightness of the stars represents the amplitude of different frequencies. The lower frequencies are near the centre.

The classic image of Lenna, which I have used here, is 512×512 pixels. The fundamental frequency of the DCT is such that half a wavelength spans 512 pixels both vertically and horizontally; the amplitudes of these “signals” are placed nearly at the centre of the graphic. The highest frequency is such that half a wavelength is just 1 pixel wide; the amplitude of these components is placed on the perimeter of a circle surrounding this centre point.

Generally, if the photograph consists of vertical lines, then the “stars” will fall in a horizontal line. The Lenna picture has lines running from the bottom left to the top right, mainly her hat. This is why there is a band of stars running from top left to bottom right.

Reconstruction

My second picture of Lenna is 256×256 pixels (Picture 7). It is a reconstruction from the DCT but omitting the highest-frequency components. The graphic on its right is a 3D representation of the amplitudes of the frequency components making up the 2D DCT.

Although you might not realize it when you look at the full 256×256 reconstruction, the picture is very slightly, almost unnoticeably, blurred when compared with the original 512×512 pixels. However, if you look at the

blown-up version of a detail taken from each of the two pictures, you will be able to see this difference.

Picture 9 is clipped from the original (512×512 pixels); Picture 10 is clipped from the reconstructed picture (256×256 pixels).

The reason that we don't notice these genuine differences when we look at the full pictures is that whatever part of our brains we use to process the pictures fills in the jagged bits with the smooth lines we expect; we experience this as "slight blurring". The reduction in resolution and file size is no illusion; the illusion, that the two pictures are virtually the same, takes place within our brains.

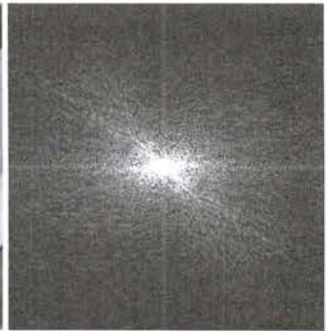
Gerald Fitton

gerald@archivemag.co.uk

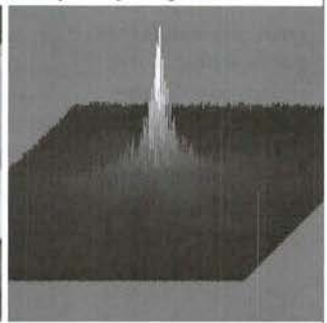
Note change of email address.



Gerald is aware that we are all stardust – stardust which is under the illusion that it is alive. However, as Joni sings, "it's life's illusions" that he experiences daily. Gerald is under no illusion that "life's illusions" is the only reality we have!



(5) Lenna – well, a 512-pixel square of her – and (6) a pictorial representation of the Discrete Cosine Transform of this photo.



(7) Lenna reconstructed to 256 pixels square. (8) A 3D graphic showing amplitudes of the frequency components of the 2D DCT.



(9) A detail clipped from picture 5; (10) same detail clipped from 7.

Lena: 'First Lady of the internet'

In 1973 a group of engineers at the University of California Signal and Image Processing Institute needed a digitized image for a conference paper. The standard test images that were around at that time were television test pages with the usual sharp-edged lines and circles; for their presentation they wanted something eye-catching with graduated shading.

They found Miss November, "Lenna", in a recent issue of *Playboy*: a full-length picture of Lena Söderberg.

Their scanner was a Muirhead wirephoto machine – which since the 1950s was in common use by newspapers to send pictures around the world by wire. It had a cylindrical drum and its fixed resolution was 100 lines per inch.

The California engineers wanted a 512×512 image, so they used the top 5.12 inches of Lenna. They



Muirhead fax machines, like this one with a cylindrical scanning drum, were widely used since 1950 to transmit newspaper photos by wire. The young engineers in 1973 added analog-to-digital converters to it, one each for red, green and blue.



IS&T's 50th Annual Conference

Lena Söderberg and the image expert who brought her to the 1997 conference. She had been unaware of her decades of digital fame until a computer magazine tracked her down in Sweden.

Because the Playboy article had changed the spelling of her name, the familiar test photo from 1972 is often known as "Lenna".

wrapped the cropped picture showing just her head and shoulders around the drum of the scanner and created the first digitized picture of the now famous Lenna.

In 1997 Lena was a guest speaker at the 50th annual conference of the Society for Imaging Science and Technology; she gave a presentation about herself. She lives in Sweden with her husband and three children. Today her charity work includes helping handicapped people use computers. If you google "First Lady of the internet" you will find that this is Lena Söderberg.

Of course the original full-length image of "Lenna" on the centrefold of the November 1972 *Playboy* (its best-selling issue – more than 7 million copies) can be found easily on the internet but, rather than include it here in Archive, for my second picture of Lena I chose one from the May 1997 conference at Boston.

The secret life of algorithms **Part 9**

Nicholas Cutler

The algorithm I wish to examine in this article is another that most people will recognize as being in daily use – and most Archive readers will probably have wondered how it works. It is also a procedure that neatly draws on concepts that I have introduced in some previous parts of this series.

The broad purpose of the Jpeg (Joint Photographic Experts Group) routine will be familiar enough: to compress an image. As an increasing number of people take advantage of the facilities offered by digital cameras there is an increasing need to store large images in the minimum space.

The common compression methods I looked at in part 4 (*Archive 22:12*) are simply not very effective when applied to photographic images. First, photo files are just too large – much greater compression ratios are called for. Secondly, typical photographs do not possess the long runs of repeating patterns that most compression methods rely on.

Jpeg differences

Where Jpeg differs from other compression algorithms, however, is that it is a “lossy” format; after decompression the image is not exactly the same as the original. Very often this doesn’t matter: for example, in a continuous-tone image you would need to look very closely to tell the difference between two adjacent shades of grey – although, as we shall see, this can be a problem in some cases with large areas of uniform colour, or sharp contrasts between adjacent shades.



Jpeg compression

Before I start to explain the algorithm I should mention that, in fact, Jpeg is a broad term for many different methods of compressing images, including a lossless format, and even another for two-colour (black-and-white) images. The particular method that I will write about as the Jpeg algorithm is in fact the original one, and is more properly called **baseline Jpeg compression**.

Two-dimensional Fourier

The principal feature of this compression algorithm is the application of a method related to the Fourier transform which I covered back in part 5 (*Archive 23:2*). The main difference is that in this case it is applied in two dimensions.

Intuitively, at least, this seems a strange idea: what does a method for analysing frequencies of periodic waves have to do with images? Start, therefore, by considering a two-dimensional wave pattern, like ripples on the surface of water or oscillations on a drum skin. Now imagine that these waves are coloured according to height: the peaks are white and the troughs black. Alternatively, visualize the sample values of a digital audio file as points of a one-dimensional image coloured the same way.

These two examples would perhaps give rather contrived images, but they do demonstrate the concept. For a more natural example, consider the pattern of shadow falling on the ground. The photograph of such a scene shows first light, then dark, and finally light again. Clearly, therefore, images can be said to

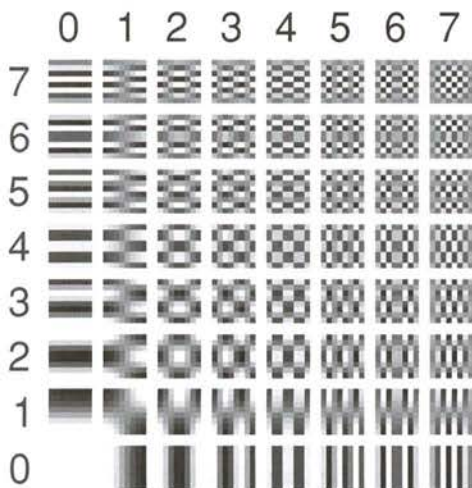
have frequency components!

The first step in applying this is to split the original image up into square blocks, each of 8×8 pixels. To each of these we apply the Discrete Cosine Transform (DCT), to obtain the frequency components in both dimensions. The equation shows how this procedure is written mathematically.

The similarity with the Fourier transform should be apparent, but here the summation is taken over both the horizontal and vertical dimensions. Using notation similar to that of the Fourier transform, $F(i, j)$ denotes the frequency components, and f_{mn} is the brightness of a pixel in the original image block. For now, to simplify this explanation, I am assuming that the image is greyscale at 8 bits per pixel. The coefficients C_k take the value $1/\sqrt{2}$ when $k=0$ and 1 otherwise.

Think of a set of matrices

Another way of viewing this



The 64 basis matrices, each 8×8 bits.

$$F(i, j) = \frac{C_i C_j}{4} \sum_{m=0}^7 \sum_{n=0}^7 f_{mn} \cos \frac{(2m+1)i\pi}{2N} \cos \frac{(2n+1)j\pi}{2N}$$

The Discrete Cosine Transform doing a summation in two dimensions.

transformation, if the idea of considering images in the frequency domain still seems odd, is to think of the $F(i, j)$ coefficients as those of a series of 8×8 basis matrices. The weighted sum of all 64 of these basis elements gives the original block from the image. The basis functions range from a block of uniform white, through striped patterns to one like a chessboard. These are shown at the bottom of the page.

Again, to confirm the analogy, each of these patterns can also be seen as a two-dimensional wave, where the height is equivalent to the brightness.

Let Lenna help

Having applied the DCT to each of the 8×8 blocks from the original image, we end up with the blocks of transformed coefficients, as explained above. At first this doesn't appear to have helped: the result is a matrix of floating-point numbers, where the original had merely 8-bit integers. This has hardly represented the image any more efficiently!

If you wish to follow this on a real example, I have used an 8×8 block taken at random from a real greyscale photograph (the famous Lenna image used in almost all Jpeg examples). The transformed matrix is shown on the next page and also in the file called [Files: see p.2](#) Example provided on the Archive website to go along with this article.

Look more closely at the relative magnitude of the transformed coefficients and notice that the largest values all

An example of Jpeg compression

138	132	133	142	133	125	120	114
132	132	131	134	119	117	110	103
142	136	124	130	110	103	101	103
146	145	127	123	100	105	100	101
144	139	123	114	100	95	97	98
139	124	105	104	91	93	100	105
125	112	113	106	92	101	110	111
128	116	115	107	102	104	110	130

(1) The original greyscale values of the pixels in an 8x8 block.

-91	47.47	17.6	-4.24	18.5	2.02	0.02	0.22
83.81	16.88	-39.94	-0.44	3.32	-2.91	0.56	1.17
29.56	-36.17	-12.45	0.17	0.25	0.62	6.86	-3.9
-6.11	4.94	-1.03	6.8	-1.85	0.82	2.02	-1.9
8.5	-0.78	4.34	-0.54	0	-1.68	9.84	1.98
10.92	0.76	2.2	2.56	-5.38	4.6	-2.6	1.17
-1.15	-5.71	5.86	-2.74	1.25	0.47	-2.55	-0.76
-11.34	-4.1	6.12	2.68	1.11	-2.29	-0.63	3.72

(2) After the Discrete Cosine Transform, note the dominant low-frequency coefficients in the top left.

-6	4	2	0	1	0	0	0
7	1	-3	0	0	0	0	0
2	-3	-1	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

(3) The same after quantization and rounding to the nearest integer.

(	-6	,	0	)
(	4	,	0	)
(	7	,	0	)
(	2	,	0	)
(	1	,	0	)
(	2	,	1	)
(	-3	,	0	)
(	-3	,	3	)
(	-1	,	1	)
(	1	,	EOB	)

(4) After the diagonal reordering, nearly all of the non-zero values have come together at the beginning of the sequence, and all the zeros at the end where they don't need to be explicitly written.

130	133	137	139	135	128	119	114
129	130	130	127	122	115	107	103
137	133	126	118	110	104	99	96
148	140	128	115	106	100	98	98
147	138	123	109	99	96	98	100
135	126	112	99	92	93	97	101
126	119	108	98	95	98	105	110
126	120	111	105	104	109	116	122

(5) Restoring the image starting with the quantized coefficients, these are the greyscale values of the decompressed image.

congregate toward the top left of the matrix. This is because in most photographic images the colour or brightness does not change rapidly, meaning that most of the detail can be represented in the low-frequency components.

This means that it is actually possible to ignore the high-frequency components altogether without losing too much detail from the image. This is where the lossy nature of Jpeg compression comes in: we save storage space by discarding the relatively small high-frequency components. Similarly, there is also an element of choice, and if higher quality is required more of the high-frequency components can be retained.

Scaling down the frequencies

This step is achieved by scaling all the frequency components down, and then rounding them to the nearest integer. The scale factors are specified in the Jpeg standard and were determined experimentally. The important point to note is that the high-frequency components are scaled by a greater amount, so that, after being rounded, many of them are equal to zero.

These scale factors are specified as the elements of another 8x8 array, denoted by **Q** for the quantization matrix. In this way the process of scaling and rounding the frequency components can be specified mathematically as

$$F'_{ij} = \lfloor F_{ij} / Q_{ij} + 1/2 \rfloor$$

[The strange brackets, usually known as the "floor" function, mean round

this quantity down to an integer. —Ed.]

To control the overall quality (and therefore compression ratio) of the image, all of the entries of the matrix **Q** are made proportionally smaller to increase the image quality, or larger to reduce it. These two steps, transforming the block and quantizing the resultant coefficients, can be performed in Basic using the following function:

```
DEFNDCt(i%,j%)
LOCAL m%,n%
sum=0
FOR n%=0 TO 7
  cn=COS((2*n%+1)*i%*PI/16)
  FOR m%=0 TO 7
    cm=COS((2*m%+1)*j%*PI/16)
    grey=(img%(m%+n%*8))-128
    sum+=(grey*cm*cn)
  NEXT : NEXT
IF i%=0 THEN sum=sum*SQR(1/2)
IF j%=0 THEN sum=sum*SQR(1/2)
=INT(sum*0.25/Q(i%,j%)+0.5)
```

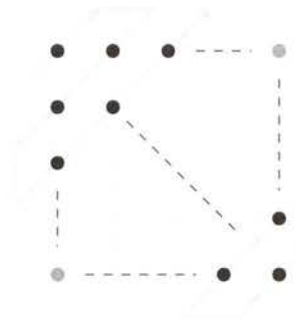
In this function the work of calculating the DCT is carried out in the nested **FOR** loops. The implementation here is rather naive, closely following the mathematical definition. As this does not need many different values of the cosine function, it would be possible to save time by calculating them all in advance and looking them up when needed. There are also recursive formulations, similar to the FFT algorithm in part 5 (*Archive 23:2*), which can further speed up this procedure.

A long run of zeros

Ultimately, this is much more helpful as the long runs of zero values in the transformed and quantized matrix do not need to be saved directly. Rather, remembering the idea of Run Length Encoding (see part 4 of this series, *Archive*

22:12), all we need to save is the value of each coefficient followed by the number of zero values. After the last non-zero entry, a special value is written to indicate “end of block”.

In this process, the two-dimensional array is implicitly being viewed as a one-dimensional series of the transformed component values. The natural way of doing this is to read through the matrix from left to right, top to bottom. However, remembering that the bulk of the non-zero values are in the top left, it is more helpful to scan diagonally, alternately working up, to the right and down, to the left. This



results in the long runs of zero values coming together toward the end where they can be more effectively compressed. This diagonal scan starts with the top-left coefficient and proceeds along all the diagonals of the matrix like this:

The result of the two preceding steps is to represent the quantized coefficients as a linear sequence of pairs of numbers (as shown on page 41 and in the example file). In each pair, the first entry is the value of the (non-zero) coefficient, and the second is the number of zero coefficients that follow it. The last pair includes the special “end of block” value indicating that there are no further non-zero coefficients.

Diagonally in Basic

This diagonal scan technique and run-length encoding of the coefficients is performed in Basic by this procedure:

```
DEFPROCdiagscan
LOCAL I%,e%,m%,n%
ZC%=0 : coeff=FT(0,0)
PRINT "("coeff", ";
FOR I%=1 TO 63
READ e%
m%=e% MOD 8 : n%=e% DIV 8
coeff=FT(m%,n%)
IF coeff<>0 THEN
PRINT ZC%)" "
ZC%=0
PRINT "("coeff", ";
ELSE
ZC%+=1 [Short for ZC%=ZC%+1. —Ed.]
ENDIF
NEXT
IF ZC%<>0 THEN PRINT "EOB)" ELSE PRINT "0)"
ENDPROC
```

In this procedure the element numbers are read from a series of DATA statements (included in the full program) before being converted to the indexes of the two-dimensional array as required. If the coefficient in question is zero then it is sufficient only to increment the “run-count”. For demonstration purposes the pairs of coefficient and run-count are printed to the screen, but in reality they would be stored in memory for further processing.

One further refinement

For many typical photographic images this gives a much more compact representation than simply storing the brightness value of each individual pixel. Essentially, we have utilized the DCT to change the original data to the frequency domain, and then

applied fairly simple, conventional techniques to encode the result more efficiently. There is, however, one further refinement used in Jpeg compression.

Recall from the example above that there were a small number of relatively large values in the top-left corner, a few much smaller values, and finally many zeros in the bottom-right. By studying these values from a large number of typical images it will be discerned that certain values are more common than others. Considering the pairs of coefficient and run-length, we can see, even from the simple example here, that large values followed by short zero counts are relatively rare. On the other hand, small coefficient values with long runs of zeros are much more common. By using Huffman coding (again mentioned in part 4, *Archive 22:12*), the more frequent values are given short code words resulting in a further saving in storage space.

Decompression

Decompressing the image performs the reverse of all the above steps. Starting with the Huffman-coded pairs of coefficient and zero count, we proceed to initialize the quantized array, again using the technique of scanning diagonally. Reversing the quantization is simply a question of scaling all the values up by the relevant factor. At this point note that it is impossible to completely restore the coefficients that were lost in the rounding process.

Finally, the inverse of the DCT is performed to give a close approximation to the original image block. As with the Fourier transform, the inverse of the DCT is almost identical to the definition above, but the pixel values are replaced with the re-scaled, transformed coefficients. The details can be found in the function

FNinvdct [in the Basic file that you can download from the Archive website].

And now add colour

With all of the details in place for the compression of greyscale images, it is now possible to examine how the Jpeg algorithm handles colour. First, recall that the colour images from a digital camera or as displayed on a computer monitor represent colour with red, green and blue components. This RGB colour model is the natural choice when using devices like a monitor.

It would, of course, be quite possible to treat each of the three colour components separately using the techniques described above. This technique, however, does not give the best possible compression performance, because there is often considerable correlation between the three components: as one value changes, so do the others.

A different colourspace

To solve this problem the image is converted to use a different colour model in which each pixel is specified by a brightness value (*luminance*) and two colour (*chrominance*) coordinates. This is the so-called $YCbCr$ colour model, which is related to the more familiar RGB space by a set of linear equations:

$$\begin{pmatrix} Y \\ C_b \\ C_r \end{pmatrix} = \begin{pmatrix} 0.29 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.5 \\ 0.5 & -0.419 & -0.081 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

Beyond reducing the correlation between the luminance and chrominance values, this colourspace often results in considerable redundancy in the two colour coordinates. This means that these can often be compressed more efficiently by exploiting the fact that they change only

slowly across the image. Indeed, these colour components are usually sub-sampled by discarding every other row or column before applying the core algorithm to the C_b and C_r components separately. When decompressing the image, adjacent rows are then duplicated to give the original number of values. As with quantizing the DCT coefficients, we are deliberately discarding some of the data to obtain better compression.

Compression at the cost of detail

It is important to understand that the Jpeg compression algorithm offers higher compression ratios at the cost of removing some detail from the image.

With typical photographs, where adjacent pixels do not exhibit marked changes in brightness or colour this is often acceptable. Most of the detail is in the low-frequency components, so that those high-frequency ones that are discarded contribute very little. Furthermore, in a continuous-tone image, the eye is not overly sensitive to occasional pixels that differ only slightly from their original values.

Not so good for screenshots

However, if the algorithm is applied to images with uniform blocks of colour and sharp contrasts between features, then the performance is less favourable. For example, consider saving a screenshot of the Risc OS desktop in Jpeg format. If you have tried this you will often have noticed fuzzy areas around text and the edges of windows.

This is because, unlike photographic images, most of the detail is in the high-frequency components. (The one-dimensional Fourier transform exhibits a similar problem with square

Where Jpeg's flaws show up



An 8x8 block of pixels taken from two different images: from the Lenna photo (above) and from the corner of a black-and-white icon (below).

At right, the shaded pixels indicate how widely the Jpeg compression differs from the original.



black-and-white image (intended to represent the corner of an icon). Notice how the errors are much larger in the latter case despite using the same “quality” setting in both cases.

Like the general-purpose compression methods introduced in part 4, Jpeg compression needs to be used with caution. If the image you are working with is not a photograph, then other formats such as Tiff and PNG (both of which can offer lossless compression) are more appropriate. They may give better compression, and they will certainly give higher quality.

Files to download

For those wishing to experiment with the program, have included two example 8x8 greyscale blocks with the filenames *Imagedata* and *Lenna* among the material downloadable from the Archive website.



Nicholas Cutler ncc25@cam.ac.uk

waves.) The only solution to this is to increase the “quality” setting and to accept less compression.

This is illustrated above on two quite different blocks. The example in the top row shows an 8x8 block taken, once again, from the Lenna image. On the right, the pixels have been shaded according to the difference between the original image and the Jpeg-compressed version. The bottom row does the same with a very simple



At the training camp for rowers earlier this year, here's Nicholas replacing a faulty component on someone else's boat.

Ed's eye spies: Where does our Jpeg file icon come from?



Now I'm curious about the origin of our distinctive “eye” icon representing Jpeg files. It seems unique to Risc os (though Ro 5 has a different icon set by Richard Hallas). Google Images turns up nothing like it from other platforms. The earliest date I can find on a !Sprites file containing this icon (file_c85) is inside !ImageFS, dated 1992-06-27, which is before Acorn introduced native Jpeg support. That could mean that Majid Anwar, author of ImageFS, designed our icon.



It's similar to the application icon that Adobe Photoshop used from 1991 (v.2) to 2000 (v.6), shown at left, but the Acorn 1992 sprite looks substantially better and is larger.

My theory is that both allude to Lenna's eye, enlarged on page 39 and here:



Mac Matters

Mark Stephens

Lots of ground to cover this time round. But first a quick thank-you to the organizers of the London Risc os show. Last year's was really exciting and even a bit too crowded. Thanks for all that hard work, and I hope the 2013 show is even better.

R-Comp and CJE Micros have complete, polished systems if you want to just run Risc os built around the superfast Pandaboard, or you can buy your own kit and experiment with different operating systems. If you are remotely interested in using Risc os software, the Raspberry Pi at £30 is a real no-brainer as a fast platform to run it on, and the Pandaboard is superfast. Fall in love with computing and Risc os all over again!

New OS software – Mac and Linux

OS X is the operating system on the Mac. My version for the past year has been 10.8, “Mountain Lion”. [Apple is running out of big cats as names for OS versions. —Ed.] Unlike Windows releases, Mac updates tend to be a yearly event, carry a much smaller price tag and bring incremental improvements. 10.8's focus is very much on “the Cloud”, encouraging you to store your files online. It did not break any of my software, including Virtual Acorn.

In the world of Linux, several different distributions are on offer. A huge number of choices exist in the Linux world, and a



This is a Mac pretending to be an Acorn RiscPC. Mac OS X is running VMware, which is running Ubuntu, which is running RPCemu.

“distribution” is a different selection of software maintained by a company or even individual enthusiasts, geared at different target groups. Some people want all the latest features, some want a stable platform, and so forth.

On my Mac, I use the Linux distribution put together by Ubuntu (thanks to a program called VMware which lets my Mac run Windows and Linux operating systems as well as the Mac's OS X). Ubuntu aims to put out two releases a year, in April and October. Ubuntu 12.10 added no big new features but sorted out a lot of rough edges in 11.04, which brought in a new windowing system called Unity.

More on RPCemu

In my previous column (*Archive 23:5*) I had started playing with RPCemu. I have been running this RiscPC emulator both on OS X and under Linux on my Mac. An RPCemu stand was at the London show

2012 and is booked again for 2013. The software is looking increasingly polished. You can download it directly from the internet or buy a USB stick from Rool containing ready-to-run versions of RPCemu for both Windows and Mac.

Running RPCemu on Linux is more complicated because you have to “compile from source”. But it comes with clear instructions and is not nearly as daunting as it sounds. It means running a command to make the source code into a program and satisfying dependencies (in other words, making sure the other software that RPCemu needs is installed; you are told if something is missing). See my screenshot on the previous page.

Organizing files

With lots of versions of Risc os around on my Mac, it has become a pain organizing my files, especially when I get back from the London show with lots of new stuff and want to update Techwriter, Artworks and so on.

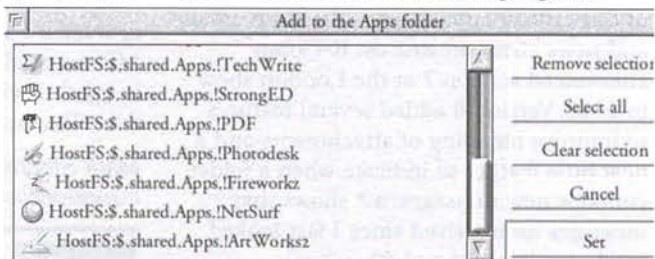
So I created a folder called shared and inside that I have a set of folders (Apps, Devtools, files, etc). I put all my programs or files there and keep Apps for the OS-specific programs such as !Paint and !Draw. Then I use a command-line tool

```
ln -s
```

which I have mentioned before. This creates a “symbolic link” – I can make one file or directory appear in lots of different places, although there is only one actual copy. I use this to make my shared folder appear in all my Risc os setups on my Mac and also put in links to several Mac locations, such as downloads and Desktop.

You can even add links to the other Risc os setups.

So when I updated all my software in RPCemu on Linux, it was also all updated when I decided to use Virtual Acorn on the Mac. It also means you can easily share all your files and programs between Risc os 5 and 6. You can still make the programs



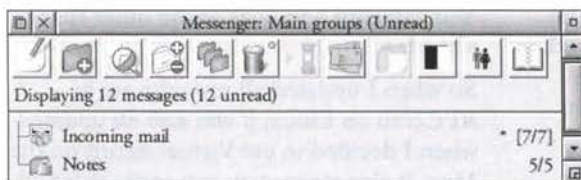
appear in Apps by using Add to apps in the Configure menu).

One word of warning: This sharing of files works very nicely (and remember that the Mac will automatically back up all your files if you have Time Machine set up). It also makes it very easy to sync to my other Risc os machines. *But* you must make sure you have only one version open at a time. There is no protection if you were to try to edit the same file on different setups. It could be corrupted or you could lose work.

Fontastic

I needed some new fonts for a project so dug out my copy of the *Professional Typography 2* CD from the Electronic Font Foundry (EFF). I had forgotten what a real gem this is. The CD contains not only 800 fonts for Risc os [including improved versions of the 500 on CD1 —Ed.] but also versions that work on Windows and my Mac as well. My daughter is especially fond of Malinka the Cat.

If you have a copy of the CD, dig it out. And if you don't, check it out: eff.co.uk.



New release of iMessage

It is great to see continued developments of this critical application – the main email and news client on Risc os. R-Comp announced version 7 at the London show in 2012. Version 6 added several features to improve handling of attachments and a neat little feature to indicate when a folder contains new messages: a * shows that messages have arrived since I last looked inside my Incoming mail directory.

Notice the Notes folder in my Messenger inbox. This actually contains stuff that I typed into the Notes app on my Iphone and then synchronized to my mail

account. It is all beautifully cross-platform.

Endnote

I have run Risc os only on my Mac for the last two years, so it has been a pleasant surprise to be using some “real” hardware again. I was so impressed with the new developments that I also ordered a Pandaboard to play with (CJE Micros, as always, was knowledgeable and helpful). I will let you what it is like next time. Exciting times on and off the Mac!

Mark Stephens

markstephens@idrsolutions.com



Mark was most impressed on his recent trip to Silicon Valley to find Google had technical documents pinned up in the Gents (he can't comment on the Ladies). Waste no opportunity to learn new ideas ...

PC Bits: at the support desk

Bimal Jangra

It's been a while again since my last instalment (*Archive 23:5*) and, as always with the world of technology, things have been changing. It seems to me as if the speed of those changes has slowed – or it might just be that things are actually going so fast that they have passed me by!

My opinion of Windows 8 stands

The first major update for Windows 8 is expected this month. Apparently 8.1 is going to put right some of the several criticisms of 8, whose uptake has been lower than any of Microsoft's previous incarnations – significantly lower than Windows 7. I think this shows the target market for Windows 8 is poorly defined.

Is it for PC users? Well, not really – unless

they have a touchscreen as well. Is it for smartphone users? Well, not really – unless you have a machine to run it on that is sufficiently portable, which means a smaller screen and so less usability if trying to do the usual PC tasks such as wordprocessed documents or spreadsheets.

So who *is* it for? My recommendations remain the same as in my previous column when I previewed Windows 8:

- If you have Windows 7, keep it and don't change.
- If you have an older version of Windows (such as XP or Vista), upgrade to Windows 7 – you won't be sorry.
- If you have legacy software that will run only on Windows XP, get Windows 7 Professional (or better), because it comes

with XP Mode, which allows you to have a virtual XP machine that does run just about everything, and it integrates seamlessly into the Windows 7 start menu – the screenshot shows what I mean.

- If you buy a new machine, check whether you can get it with Windows 7 – there are still machines being sold with Windows 7, but fewer all the time.
- If your new machine can be had only with Windows 8, buy a separate copy of Windows 7 – they can still be bought, but are running out – and rebuild the machine with Windows 7. Does that sound like a lot of work? Well, it'll be worth it, trust me!

I'm in pastures new

I left the teaching profession once and for all back at the end of 2011 – suffice it to say that I have not been pleased with the changes that have been brought into the education field over the past several years, so I voted with my feet. I started working for Capita IT Services in December 2011, and I provide IT technical support for a range of government organizations.

Obviously a lot of the job is top secret: I'd have to kill you if I told you about it. But here are some general observations about the type of work I have done for 18 months.

Which Windows?

One of the first things to point out is that, until recently, all my clients used Windows XP – in fact, my main PC is still running XP Service Pack 3. Most clients are changing over to Windows 7; some have already changed over completely. We don't encounter any other versions of Windows, and that suits me fine.

The PC specifications are generally OK – nothing is remotely cutting-edge but most machines have dual-core processors. We



The Windows 7 start menu provides an XP mode.

look after a fair number of laptops and also deal with thin-client machines, such as the Wyse T50 [overleaf]. At the sites I deal with, these thin-clients connect to servers that run Citrix or VMware.

Behind the scenes at Support

I am working as a service-desk analyst, providing first-line support. Years ago when I worked for BT, I did second- and third-line support. Is this jargon new to you? First-line support people generally log problems on a system, which can then be passed on to second-line to be fixed.

The first-line people tend to be less technical than second- or third-line support, but the depth of knowledge can vary vastly. At its most basic level, first-line support analysts merely “log and flog” the telephone calls (oh yeah: the problems are always arrive by phone). Personally, I always try to fix the issues that come to me, and I'm pleased to report that my

“first-time fix” rate averages around 70%: this means that I generally fix 70% of problems without the caller needing to be contacted by anyone else.

We have a range of remote-access tools at our disposal, which allow us to fix a range of things. We often get password resets, which are usually fixed by access to Active Directory or occasionally Novell Console One. I couldn't function without the access to AD: this is fundamental for 90% of the things we do.

Second-line support analysts have access to a greater range of support tools and generally have greater access rights – this can be especially frustrating if I know how to fix a problem but know that I don't have the access required. Third-line support engineers are generally based onsite, and can actually go in person to fix issues. This is one of the aspects of my time with BT that I especially enjoyed, and I quickly built up a reputation as being one of the people to go to in order to get all manner of issues put right!

In order to access the servers that we need to access on a day-to-day basis, we use RDP sessions – Remote Desktop Protocol – which accesses a remote session hosted by the server concerned. This is the next best thing to being able to log into the server directly, and allows us access to Active Directory and other snap-ins such as the Exchange Management Console, used for managing email accounts.

As you might guess, all of these accesses require dedicated accounts, complete with user IDs and passwords. After a while, I



A Wyse T50 thin-client computer as used by some of Bimal's callers. It measures 7×4¼×1 inches – like a Terry Pratchett paperback – and sells for around £210. Basically it is a Marvell Armada 510 system-on-a-chip, which is an Arm v7 processor plus hardware for 3D graphics, high-definition video, USB, ethernet and all the rest.

(Isn't this what Oracle was keen for Acorn to build back in the 1990s?)

found that my memory couldn't cope with the number of passwords and usernames I needed (I'm getting old!): I now need to remember more than 24 passwords. Unsurprisingly, resetting passwords and unlocking user-accounts form a significant part of the calls we get, especially after holidays. I always like such calls, as I know that I can easily put the problem right and get the caller working again.

Telephone calls galore

I've arrived at some conclusions after taking hundreds and hundreds of calls in this job. Some callers are wonderful – warm, friendly and always grateful for the help you give them. Others are impatient to get on with their work and can't bear to be held up any longer – even though it takes time to access AD, find their account, check it's not locked (unlock it if it is) and type in the new password (twice).

Some carry on talking while you're trying to ask them crucial questions in order to help them. Back at school I would just tell a pupil to listen to what I am asking, but in this job we have to remain courteous – especially as all calls are recorded and we are graded on our telephone manner.

And then there are the people who know what needs to be done better than you do. You explain how to fix an issue and they reply that they've never done that before. If they know so much, how have they got into the mess in the first place? They remind me of pupils I used to teach: I would tell them to access a website, give them the URL, and the pupils would say, "Sir, sir, it doesn't work!"; I'd say that it was because they hadn't typed the URL correctly; they'd say that they had; I'd go over to them: it was just as I said.

Sometimes things work better if people just do what I ask them to do; then everyone's happy. Ah well – it sounds as if I've come full circle, with the users reminding me of pupils at school. There's probably some mystic significance but I'm blessed if I can see what.

So please bear in mind that when you ring up for technical support (I know most of you are computer experts so you don't need to do that!) you might come through to one of my colleagues – you might even come through to me, but I can't tell you who I support.

The lovely late shift

I have high-enough security clearance to work on all of the contracts we support. I am currently one of only six such individuals, which means that we get the honour of covering the *late shift*, which runs from noon until 8pm for a week. This comes round every six weeks. When a call comes in during tGahis time, the person on the late shift is the only person available to speak to – this often pushes my knowledge to the limits since we can be dealing with a contract that we do not normally deal with on a day-to-day basis.

As you can tell, our support is based in

Britain. There had been talk of outsourcing some contracts to other parts of the world, but I'm thankful that for us this did not happen: it would have been an impractical commute!

Old 'industry standard' cliché

Doing this job has given me a greater understanding of the huge range of services and different types of software in use that no one is even aware of, since it all happens behind the scenes. One thing is very clear to me, given my background in education: there is far more to working in IT than Microsoft Office.

Although Windows is the dominant framework operating system, so many other types of software hang off it that education's old chestnut of "*It'll give them experience of what they'll use in industry*" is further from the truth than ever before.

Virtually all fields have their own dedicated software, designed to let them do the work they need to do. Yes, Microsoft Office comes along, but it is a very small part compared to the overall whole. So we need to train our youngsters to understand the general principles of IT rather than just how to drive specific consumer packages. When they get out into the big, wide world, they will need to use things they will never have seen or heard of.

Bimal Jangra hermits@btinternet.com



Bimal's band, Beyond Words, is already booked for New Year's Eve 2015. It also won an award earlier this year,

competing with a number of select acts. Beyond Words never expected to win, being the complete underdog, so the result was a thrill. As you can see, the band keeps Bimal out of mischief (mostly).

Compiling with GCC 4.1.2 Part 2

Rob Johnson

This time I'll look at the size overheads for various kinds of program in the new version of the GCC system.

But the first three sections will be about things I discussed last time. Having had more time to investigate and think, I realize that I made some hasty statements that need correction or expansion.

Floating-point processing

I talked in Part 1 (*Archive 23:6*) about the choice between hard-float and soft-float for floating-point arithmetic, between a hardware implementation and a software emulation. Unixlib has its own software routines that can be called to do FP calculations. But some Arm systems have hardware that accepts FP machine-code instructions. I claimed that requesting hard-float would create or allow such instructions. But my StrongArm does not have FP hardware. So I claimed that the Shared C Library (SCLib) detects this, and reroutes the FP instructions to its own software alternatives.

My claims weren't quite true. For a start I had forgotten that all (I think) Risc os operating systems have an FP emulation module built-in. It is this that is used if there is no FP hardware. The Shared C Library isn't involved in this choice. In fact I believe that the FP emulator always gets first look at FP instructions, and that *it* will detect the presence of FP hardware, and pass the instructions on.



GCC is still an application for compiling programs written in C, C++ and assembler. It is a free system developed on Linux, and ported over to Risc os.

Rob here continues his detailed examination of the latest release.

Where the libraries get involved is that floating-point arithmetic is used in (for instance) FP trigonometric functions. These functions in SCLib will use FP instructions. These functions in Unixlib will call the Unixlib FP functions.

The GCC system is affected because it has to choose between SCLib and Unixlib (via `-mlibsc` or `-munixlib` parameters). Also it has to decide whether to generate FP code from C or C++ as FP instructions or as calls to Unixlib FP routines (via `-mhard-float` or `-msoft-float`).

This may help to explain why 4.1.2 has dropped the Unixlib/hard-float combination, leaving only SCLib/hard-float and Unixlib/soft-float. These last two are consistent in their choice of FP implementation, whereas the other one isn't without costly changes to the Unixlib library.

All object files generated by GCC have a bit setting to say whether they are hard- or soft-float, and this applies also to the pre-created object libraries and object files in the GCC release. The linker insists that all objects to be linked together have the same float status.

The internal assembler as is part of this. It understands the standard parameters `-mhard-float` and `-msoft-float`. These set the object code to the relevant type. And also `-msoft-float` won't accept FP instructions in any assembler source you supply.

The same applies to the freestanding

Acorn-style assembler `asasm`, except that it uses parameters `-hard-float` and `-soft-float` (without the `m` prefix).

Merger of SCLib and Unixlib

Last time I mentioned that a major entry in 4.1.2's `Changes` file is that they have merged SCLib into Unixlib, i.e. the object libraries `libsc` and `libunixlib` and all the associated source files. And it is true that there are no libraries left in 4.1.2 called `libsc`, and they have been replaced by `libunixlib`. (They couldn't both exist in the same directory before.)

At the time I wondered to myself how they'd managed it. For instance, where there were routines in `libsc` and `libunixlib` with the same names (and doing the same jobs differently), how is the right one picked from the merged library?

Remember that the routines in `libsc` were dummies calling the real things in the Shared C Library module, whereas the ones in `libunixlib` did the work themselves. And the `-mlibsc` and `-munixlib` parameters still want to decide between these alternatives.

It turns out that the libraries are not quite as merged as the `Changes` log suggests. Looking back when `libsc` and `libunixlib` were separate in 4.1.1 you would find a huge `libunixlib` (~7M) and a relatively tiny `libsc` (~70K). Now in 4.1.2 the Unixlib/soft-float `libunixlib` (where it existed before) is still ~7M. But the `libunixlib` replacing `libsc` for SCLib/hard-float is now only ~500K.

Looking inside the libraries I find that the SCLib version still has the dummy calls of the routines in the Shared C Library, instead of the Unixlib full routines. The rest of the increase from 70K to 500K is made up (mostly) of Unixlib functions that aren't in the Shared C Library, so an

SCLib program now has access to these. But the difference between 500K and 7M isn't solely that lots of Unixlib functions are replaced by the small dummies. There are a large number of Unixlib-only routines that haven't been copied across (and not just the functions to be used instead of FP instructions).

I'm not going to speculate about the choice of functions left behind, but I've provided some alphabetical lists of functions in various libraries. I'm also not going to mention some other things I noticed while looking in the libraries. (Readers of my earlier articles will wonder at my restraint!)

Dhrystone and Whetstone

Last time I also used the near equality of all Dhrystone timings to support my claim that all programs on my StrongArm use software rather than hardware fp. While the conclusion is true, the evidence is rubbish. I'd forgotten that the Dhrystone program specifically and deliberately doesn't test floating-point operations. For that I need the Whetstone program.

The slight difference I found between the Dhrystone timings for 4.1.2's static SCLib and Unixlib has to be down to differences between the Shared C Library and the full `libunixlib` in other functions.

When I looked back at 4.1.1, I claimed that the Dhrystone result for the then-existing Unixlib hard-float was the same as SCLib (hard-float), different from Unixlib soft-float. I think it must have been a case of seeing what I expected to see, because I thought Unixlib hard-float must be going through the same hardware check and default to software as SCLib. (This makes sense now I remember that the FP emulator is doing that. But it didn't

actually make sense then when I thought it was the Shared C Library doing it, because Unixlib, even hard-float, shouldn't have had access to the Shared C Library.) Now when I redo the 4.1.1 tests I find that Unixlib hard-float actually has the same timings as Unixlib soft-float. Which is the right answer since floating-point has nothing to do with it.

The Dhrystone program is supplied in GCC's Examples. I got the C Whetstone program from www.netlib.org/benchmark, which holds lots of benchmark programs. It compiled without problem under GCC.

When I compared timings of the two possible 4.1.2 static program versions (Unixlib soft-float and SCLib pretending to be hard-float). I found that Unixlib was more than twice as fast as SCLib. The equivalent 4.1.1 programs have the same timings as these. The 4.1.1 Unixlib hard-float option has the same time as the SCLib (hard) version, because in both cases the compiler generates FP instructions rather than calls to Unixlib functions.

The large difference may be down to more efficient coding of the Unixlib functions over the FP emulator ones. But an indeterminate amount of it may also be due to unavoidable overheads for the SCLib FP emulator system – that is, the time taken to recognize the attempted FP instructions and decide which emulator functions to use instead. And possibly a software check on whether to use hardware repeated each time, although this might be avoided by the setting of a vector. (I presume the emulator isn't *physically* checking for possible FP hardware each time.)

AIF or ELF

Now at last I can move on to something different.

So far in this new series I have mentioned the choices between SCLib and Unixlib, hard-float and soft-float, static and dynamic programs. There is also another hidden choice, between the AIF and ELF formats for program files. AIF is the Acorn standard that Risc os GCC compilers used until 4.1.1 introduced the ELF format, which is used by the Linux original version of GCC. [*The Risc os filetype is &elf – with a numeral “one” as the second hex character.* —Ed.]

Just as only certain combinations of the other choices are allowed, there are restrictions on AIF/ELF too. SCLib programs can only be AIF. Dynamic Unixlib programs can only be ELF. Static Unixlib programs are created as ELF, but can be converted to AIF by the program `elf2aif`.

The size of AIF C programs

In Part 3 of the previous series (*Archive 21:7*) I started comparing the overheads of various types of program using the minimal code

```
int main void{}
```

Now I'm going to redo that comparison, but I'm going to separate AIF from ELF, and also consider C programs apart from C++.

As well as the current 4.1.2R2 and my original tests on the early 4.1.1PR1, I have also looked at the last 4.1.1 release 4.1.1R2. I have also added the results for 3.4.6, for reasons that will become clear in the next section.

As usual with my !Compile application I get the different compilation options by

editing my
Compile_c file
within !GCC.bin.

We see in Table 1 that the Unixlib programs increased by 5–6K across 4.1.1, but then 4.1.2 has slightly shrunk the only valid option. Meanwhile the SCLib version has remained constant at 3K (to the level of accuracy I'm going into here). (The SCLib drop from 10K in 3.4.6 to 3K in 4.1.1 was explained in Part 3 as due to a rationalization of what needed to be included for every SCLib program.)

AIF C++ program sizes

Table 2 shows the figures for the same minimal code as a C++ program.

These results were obtained, as standard for my !Compile app, by using the g++ frontend.

In Part 10 of the last series (*Archive 22:4*) I mentioned that the dummy C++ program didn't actually need any of libstdc++, and the stuff loaded by default could be avoided by changing my Compile_cc file to compile with gcc rather than with g++. GCC doesn't link in libstdc++, but the cc source directory still directs it to compile as C++ rather than as C.

The point here is that this enables me to see what size C++ programs are without the C++ Standard Library. Not surprisingly the results were identical to the AIF C programs, so I haven't bothered displaying them..

(1) AIF C programs

	3.4.6	4.1.1PR1	4.1.1R2	4.1.2R2
Unixlib static soft-float AIF	----	141K	147K	145K
Unixlib static hard-float AIF	135K	136K	141K	----
SCLib static hard-float AIF	10K	3K	3K	3K

(I can even compile SCLib C++ programs this way in 4.1.1, producing 3K. For more on the subject of what I could force 4.1.1 to do with C++, see Part 10 above and some of its preceding articles. That should now be of historical interest only, as 4.1.2 has integrated C++ and SCLib again. Which I will probably check in a later article.)

A further point is that I can also see how much libstdc++ added to each program, by subtracting the C figures from the C++

(2) AIF C++ programs

	3.4.6	4.1.1PR1	4.1.1R2	4.1.2R2
Unixlib static soft-float AIF	----	195K	202K	200K
Unixlib static hard-float AIF	157K	190K	195K	----
SCLib static hard-float AIF	10K	----	----	57K

ones. The result (table 3) shows that there is an almost-constant addition of 54–55K in 4.1.1 and 4.1.2. The apparent difference is explained by the rounding applied to the individual figures.

Overall we see that the libstdc++ overhead went up from 22K in 3.4.6 to 54K in 4.1.x (which was explored in Part 6 in *Archive 21:12*). It is unchanged in 4.1.2.

(The question of why there was *no* libstdc++ overhead in 3.4.6 for SCLib was also

(3) AIF C++ programs libstdc++ stuff

	3.4.6	4.1.1PR1	4.1.1R2	4.1.2R2
Unixlib static soft-float AIF	----	54K	55K	55K
Unixlib static hard-float AIF	22K	54K	54K	----
SCLib static hard-float AIF	0K	----	----	54K

discussed in Part 6. The answer was relevant to why I can get away with dropping libstdc++ for my dummy program in 4.1.x.)

Static ELF C program sizes

ELF programs didn't exist in 3.4.6 and don't exist for SCLib. So we can look only at Unixlib programs within 4.1.n. Moreover 4.1.2 has dropped hard-float for Unixlib, so the only full comparison will be for Unixlib static soft-float programs.

The interesting comparison here is between ELF and AIF versions of the same program. Also, in Part 5 of the previous series (*Archive 21:11*) I found that the size of static ELF programs could be significantly reduced by two parameters: -s and -wl, -S.

I would like to clarify something from Part 5 which I for one found confusing when I reread it for this article. There are two kinds of excess data which these parameters remove. -wl, -S gets rid of debugging stuff. -s also gets rid of the debug stuff, plus a couple of tables of information. Putting both parameters together is actually a bad move as -wl, -S takes priority, and you lose only the debugging.

In table 4, the "ELF -s" column shows that there is still some ineradicable overhead for ELF programs. But notice that in 4.1.2 this has shrunk from 6K to 1K. This is basically down to clearing up an anomaly I noted in Part 5. The .comment section used to contain many copies of the GCC version string; now it has just one. This, along with the shrinkage in the underlying AIF program from 4.1.1R2 to 4.1.2, makes the minimal ELF program for 4.1.2 actually smaller than that for 4.1.1PR1.

Nearly all of the extra savings of -s over -wl, -S is in the symbol table .symtab. As might be expected this rises from 4.1.1PR1 to 4.1.1R2 with the increase in AIF size, and similarly shrinks from 4.1.1R2 to 4.1.2R2. In other words it depends on the amount of code included in the programs by default. But the ELF -wl, -S column shows that the 4.1.2R2 value of this element is again below the level of 4.1.1PR1. The disproportionately large drop in this table saving between 4.1.1R2 and 4.1.2 suggests some deliberate optimization has occurred in 4.1.1R2.

However, the last column shows that the debug data has suffered a continual increase, so that the 4.1.2 figure suggests that there is more debug stuff per piece of AIF code than there was in 4.1.1. Moreover this increase in 4.1.2 is enough to counteract the savings in the other paragraphs, and the total program size increases.

(4) Unixlib static soft-float ELF C programs

	AIF	ELF -s	ELF -wl,-S	ELF
4.1.1PR1	141K	147K	323K	801K
		+6K	+176K	+478K
4.1.1R2	147K	153K	337K	838K
		+6K	+184K	+501K
4.1.2R2	145K	146K	299K	876K
		+1K	+153K	+577K

Static ELF C++ program sizes

Table 5 shows that the same pattern appears with C++ programs as with C ones, amplified by the extra C++ code.

(Note that the +7K as opposed to +6K in the 4.1.1PR1 rows is more an artifact of the rounding of individual figures than it is of real code differences.)

Dynamic programs

There is no significant difference between the small dynamic C and C++ programs. C++ programs are slightly larger, but usually not enough to change the figures when rounded to the nearest K.

All fully-shrunk (-s) dynamic programs are about 3K. The unshrunk 4.1.1 programs are over 6K, and can lose nearly 1K of debugging and nearly 3K of tables. The 4.1.2 programs increase to about 13K because they have 9K of tables. The extra seems to be stuff that is included in all static programs but that in 4.1.1 wasn't deemed necessary for dynamic ones. Note that the -s parameter shows that the dynamic programs work without these tables anyway, no matter what they contain.

Files

I've provided three lists of functions. 411S shows the functions available to SCLib programs in 4.1.1. 412U has the full list for Unixlib programs in 4.1.2. 412S is the intermediate list of those functions that 4.1.2 has made usable by SCLib.

I've also included the Whetstone program and my minimal c and cc programs.

(5) Unixlib static soft-float C++ programs

	AIF	ELF -s	ELF -WI,-S	ELF
4.1.1PR1	195K	202K	404K	1033K
		+7K	+202K	+629K
4.1.1R2	202K	208K	423K	1074K
		+6K	+215K	+651K
4.1.2R2	200K	201K	387K	1116K
		+1K	+186K	+729K

Summary

This was an article of two halves. First, an expansion of last time's discussion of floating-point and the SCLib-Unixlib "merger". Second, an exploration of the changes in minimum program sizes in the new compilation system.

And next time? Who knows? Certainly not me! Maybe a look at compilation and linker times for the minimal programs, and whether I can cut them down.

Rob Johnson robjf@robjf.bbmax.co.uk



Not long ago I had a prostate cancer scare. I had a PSA blood test for the first time, and it showed my level was too high. After a biopsy showed nothing, they put me on six-monthly blood tests. The third one showed a marked PSA decrease. My doctor attributed this to my not riding my bicycle for a while (due to bad weather, old age and increasing laziness)!

Curious proofreader asks: Which apps did they use?

One of this issue's proofreaders, Chris Newman, is curious about what applications various authors use to produce their graphs and tables. Well, most arrive at Archive HQ as drawfiles, and your editor often takes them into Artworks to make curves thicker and tart up colours so that they will contrast suitably when printed in mono.

Rob Johnson uses Tablemate. Gerald Fitton uses Pipedream and exports drawfiles. Jim Lesurf feeds data from his audio-test programs into !Tau to make drawfile graphs.

"It'd be interesting for contributors to list all the Risc os apps they used," Chris said. "It might help others or trigger off possibilities they hadn't thought of. For instance, to do the proofreading, I used !PDF to read the PDF files, !Paint because !PDF didn't show the π character, Wordhound to check a spelling, !Unictrl and Lanman98 to send a file to the PC (so I could use two screens) and !Pluto to send you the results. Adds up, doesn't it!"

Charm – a practical guide

Pete Nowosad

Computers can accurately and quickly manage, manipulate and transform information in accordance with sets of coded instructions packaged into operating systems and programs. In this article of the series about the Charm language [the first part was in Archive 23:5] I am going to concentrate on how Charm defines and handles different types of data used to represent that information.

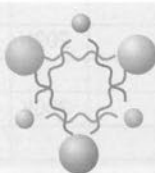
Basic data types

Currently Charm defines four intrinsic types of data as shown in the table below.

Boolean values represent flags that are either true or false, and strictly speaking only require one bit of data. It is possible to combine up to 8 boolean flags in a `char` or 32 flags in an `int` using the Charm bit-operators. For convenience and speed however, the compiler does not attempt to pack boolean definitions together but stores each in its own byte location, which is the smallest unit of memory the Arm CPU can directly address.

Keyword	Size (bytes)	Use	Example
boolean	1	Logical flags	true
char	1	Characters	'C'
int	4	Integers	-321
real	8 (d) or 12 (e)	Real numbers	1.23e-7

Part 2: Data



Pete Nowosad first invented the object-oriented Charm programming language back in the 1990s for the

Archimedes, and, inspired by the Raspberry Pi, recently recompiled and further developed it for modern Arm processors.

It is suitable for beginners yet is powerful enough for substantial projects. It has similarities to the C and Pascal languages, with unique features of its own and is characterized by “context-free grammar”. Gavin Wraith reviewed it in Archive 23:4.

Charm is free to download from charm.qu-bit.co.uk

Char values represent single characters using the ubiquitous Ascii code. Some characters are non-printing and for historical reasons are known as control characters. The most common are space (32), tab (9), linefeed (10) and carriage return (13).

Int values represent signed 32-bit numbers in the range -2147483648 to +2147483647; negative numbers have the top bit set.

When handling **real** numbers Charm will use IEEE double-precision floating-point if VFP support is enabled, or extended precision if using the standard Risc os floating-point emulator. Note that at the cost of a few trailing digits, floating-point operations complete ten times faster with VFP support, for they are performed in hardware rather than in software.

Constants

Charm allows constant values of basic types to be named using the `const` keyword. By convention the source code in the Charm distribution uses uppercase with an underscore delimiter between words for constant names. The type of the constant is inferred from the value it is assigned (using BNF notation) as follows:

- [true | false] – boolean

- '?' – char, e.g. 'Q'
- [+ | -] [0 - 9]+ – int, e.g. -123
- [+ | -] [0 - 9]+ . [0 - 9]+ [e [+ | -] [0 - 9]+] – real, e.g. 2.345e-6

The hex and bin prefixes introduce constants that are coded in hexadecimal ([0 - 9, A - F]+) or binary ([0, 1]+) number bases. For example, the space character is hex 20 or bin 100000.

Non-printing characters that are not part of the Charm language per se can be included as character data by escaping them using the '\ ' character e.g.

- \t for tab
- \n for carriage return
- \\ for backslash

Note that constants may contain expressions, and these may reference other constants as long as the result of the expression can be resolved at compile time. For example (a single line in Charm):
time. For example (a single line in Charm):

```
const ROWS = 4, COLUMNS = 6,  
CELLS = ROWS * COLUMNS;
```

Variables

A Charm variable must be declared before it can be used. In the declaration the variable is given both a *name* (conventionally lowercase with an underscore delimiter between words, and starting with a *z* for a pointer) and a *type*. Variables can be declared to be of either a basic or a more complex type. The complex data types are derived from basic types in a number of ways:

- by homogeneous aggregation using the array keyword.

- by heterogeneous aggregation using the record keyword or using module-level dynamic data declarations.
- by using the ref keyword prefix to create a reference (or pointer) to a variable of a given type.

Arrays contain data collections accessed using an index that runs from 0 to one less than the size of the array using square-bracket notation e.g. variable [index]. The index must resolve to type integer.

Records contain a collection of typed fields, each of which is given a different

name within the record-name space so that it can be accessed in the form

<record name>.<field name>,

where by convention record name is in camel case [a

MidPoint CapitalLetter—Ed].

Dynamic module data can be accessed in a similar fashion as <ref module

name>.<variable name>, where the module-name reference may be this if inside a dynamic procedure.

ref-type pointer variables are automatically dereferenced by Charm when the variable

to which they point is accessed. Variables of complex data type are always passed by reference to procedures, though they can be declared directly as static or stack-based variables. While the proc keyword introduces procedural code, it is possible to declare variables of type ref proc that point to procedures and to execute those procedures via their pointers if the correct signature is adhered to. Note that since procedures can optionally return a value of a specified type using the return keyword,



Peter Nowosad's ebook, *Charm Programming on the Raspberry Pi*, was published on Amazon's Kindle site in June, at £4.66 inc Vat. It's aimed at both children and adults; no previous programming experience is required.

in this case calls to them can be substituted in expressions for a variable of that type.

Variables of type `ref` can be assigned the special value `nil`, meaning that they are not currently pointing anywhere. When set to this value an attempt to use them to access data will result in a runtime exception; however, they can still safely be compared with the `nil` value to see whether they contain a valid reference.

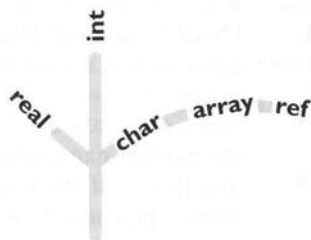
Strings

Now let's take a closer look at a particularly prevalent type within Charm, namely `ref array char` to point to a string:

```
array [] char hello = ('H', 'e', 'l',  
'l', 'o', 0);
```

```
ref array char hello_string = hello;
```

where the relevant portion of the Charm compiler type tree is:



We can output this to the screen on a line of its own using the standard `vdu` stream of the `Out` library module as follows:

```
Out.vdu.str (hello_string).nl ();
```

A few points to note:

- All strings are terminated by a zero character.
- The length of the string array is implicitly calculated by the compiler, but if it were to be explicitly entered it would be `[6]` to accommodate the terminating zero.
- Characters can be extracted from the

string using an array index starting from 0 and running to one less than the string length, for instance `hello_string[1]` contains `e`.

Since strings are used so much, the compiler associates the type of a string literal such as `"Hello"` with type `ref array char` and automatically adds the trailing zero, so more concisely we can write:

```
ref array char hello_string = "Hello";
```

The story on strings does not end there. The Charm libraries contain a `String` module that can hold a dynamic instance of a modifiable string. This is held in the heap, can grow as required, and is automatically released by the `String` destructor when the `String` object goes out of "scope" (see below).

Type conversions

In general the compiler will not allow variables of different types to be combined in assignments or expressions, since this frequently makes no sense – for instance assigning a real to a boolean. The three arithmetic types `char`, `int` and `real` can, however, be mixed, and the compiler will generate a result of the widest width (greatest number of bits) of the operands. You may also mix these types on the left and right of assignment statements, but watch out for loss of precision if the left side is narrower than the result being assigned to it.

Charm does permit a couple of powerful casting operations through the `address (variable)` construct, which can convert a variable of any type to an integer, and the `[ int ]` construct, which can convert an integer to any type. With great power comes great responsibility, however – so these are used but rarely and then with care. You can see an example in the

Charm debug library module, which casts the heap address to an array of bytes so that their contents can be dumped to a file.

Scope

The scope of a variable concerns two things: where it is accessible from and its lifecycle. A variable can reside in one of the following places:

Location	Lifetime	Construction
static	program	type v := value;
heap	variable	type v := new V;
stack	scoped	type v := value;

Variables declared *statically* exist for the lifetime of the program and are initialized at compile time, by default to zero or nil for references. Variables declared on the *heap* are created by the `new` operator and released using the `delete` operator when no longer needed. Variables on the *stack* exist until their containing scope is released. Typically this is when a procedure exits, or execution of a block of code between { and } delimiters finishes.

The specially named startup and shutdown procedures can assist with setting up static variables, and the `new` and `delete` procedures can help with setting up heap- and stack-based variables when more than a single compile or runtime assignment is required.

An important point for hiding information is that constants, variables and procedures are (by default) accessible only from within the module in which they are declared. By using the `export` keyword, however, it is possible to make these definitions publicly available to other modules. Referencing modules must explicitly state their dependencies using the `import` keyword to state each module whose public definitions they wish to use.

Such references must then be qualified with the name of the imported module. For instance, `Out.vdu.str` refers to the `str` procedure of the `vdu` record in the `Out` library used to output strings to the screen.

Next time I intend to cover some of the coding side of Charm, including writing expressions and flow of control statements.

Have a go

Finally, as an exercise, I suggest replacing the default `!NewProject MyProject` module with the following code to output square roots for the first dozen integers. The calculation uses the Newton–Raphson method, which Nicholas Cutler covered in his *Secret Life of Algorithms* in *Archive 23:5*. *[In these narrow columns, tightly packed lines indicate a single line of program code.]*

```
import lib.Out;
module MyProject
{
  proc sqrt (int n) real
  {
    real v := 1;
    for int i := 0 step inc (i) while i < 5
    v := (v + n / v) / 2;
    return v;
  }
  export proc start (ref array ref array
  char argv)
  {
    for int i := 1 step inc (i) while i <=
    12
    Out.vdu.str ("sqrt ").num_fld (i, 2).str
    (" = ").float_fld (sqrt (i), 10).nl ();
  }
}
```

 **Files:**
see p.2

Peter Nowosad pete_nowosad@msn.com



As a Java developer I enjoy drinking Taylor's Hot Lava Java coffee from my Charm mug. On a scale of one to six, it is of course strength six.

More ednotes

Armini initiation – This issue of Archive has mostly been produced on an Armini computer, on loan from R-Comp. For many years my faithful workhorse has been the Acorn RiscPC running Risc OS 4.39; all the Archive files still reside on it, and the Armini reaches them by network.

For the most part, all went smoothly. It is a joy to see pictures in deep colour on the new hardware; I could use only 256 colours on the RiscPC if I wanted a big screen. Most applications, including Impression Publisher, did not need to be installed on the Armini: they are happy to load over the network. Fonts, however, need to have their own copy.

A few niggles: SparkFS gets into a twist if I try to read a zipped folder from elsewhere on the network. Sometimes the Armini (OS 5.19) loses contact while copying a large file over the network. Organizer can be very sluggish, possibly because the Armini's storage card is slow for small files.

In search of write-ups

Various hardware and software has yet to be reviewed in Archive. If you're willing to write an article on your experience with any of these, please get in touch with the editor. Plingstore and the Panda Ro top the list.



Movember jape at London show

Two years ago I started growing a moustache – a “mo” in Australian argot –

for Movember. Forthwith I turned out to *be* a case of prostate cancer. Thanks to being caught early, radiotherapy seems to have done the trick. In the excitement I never actually got round to signing up for the month-long Movember campaign to raise awareness of men's cancer.

This year, I did sign up. The rules for the Movember whisker-growing competition say players must start from scratch. So before the end of October and before starting afresh on November 1, I'll cut off *half* of the two-year Nagelschnurrbart and raise some extra cash from folks who raise eyebrows and laugh at my half-tash.

Let's have the semi-shave at the London Risc OS show on October 26. There will be a tin on the Archive stand, or you can donate online at mobro.co/jimnagel.

The Movember charity last year raised a total of £92 million in 21 countries and, they say, sparked 2.7 billion conversations about men's health: uk.movember.com

Jim Nagel jim@archivemag.co.uk



Archive DVD or CDs

The entire contents, fully indexed, of the past 22 volumes of Archive on a DVD – as plain text, as HTML and as the original Impression files.

For users still without a DVD drive the collection is also available on a pair of CDs.

Contents are identical, except that the DVD has room for all the issues of volume 22 as PDF.

DVD22 or CD22 costs £21 – or £11 as upgrade from the previous edition.

(Unless you specifically request special despatch, upgrades will be posted with your next magazine.)

Please send cheque to “Abbey Press” at address on back cover, or pay via direct bank transfer with an email to say whether CDs or DVD – sortcode **08-92-50** (Co-op, Skelmersdale), account **7072 6592** (Abbey Press).

4D**PandaRO****A****B****C**

Pandaboard ES Mini-ITX Systems

The Fastest Ever RISC OS Computer!

Specification:

- Pandaboard ES 1024MB, 1200MHz
- Mini-ITX case (three choices)
- Custom mounting bracket and I/O shield
- Significant hardware upgrade including: Battery-backed Real Time Clock, Temperature Sensor, Power control, EEPROM, FS LED module & Reset button!
- Front power button under software control, including Soft Power-OFF!
- HDMI video output up to 1920x1200
- *Standard:* SD card with RISC OS 5 & USB pen for local storage
- *Pro:* 32GB SD card (Class 10) containing RISC OS 5 & storage
- Internal 7-port USB hub
- Ports: 2 rear USB, Ethernet, Audio (optional 2 or 4 front USB)
- PSU (internal or external depending on case)
- DVD drive (included on Pro, optional on Standard)
- SD card reader, Keyboard & Mouse. Aemulor installed.

PandaRO**£499****PandaRO Pro****£595**

Please see www.pandaro.co.uk — phone or email for more details

Upgrade your ARMiniX or DIY Pandaboard to **PandaRO** spec. £P.O.A.

Baby PandaRO

Pandaboard system in a 'Micro' metal case

**Baby PandaRO £369****Baby PandaRO Pro £405**

Please phone or email for details.



The World's largest
range of Raspberry Pi
Accessories
cjemics.co.uk/rpi

'PandaInABox'
Panda with integrated LCD
cjemics.co.uk/piab

**STOP
PRESS**

System delivery: UK £10 (Europe from £24)

sales@cjemics.co.uk

01903 523222

www.cjemics.co.uk



London Show 2013

At this exciting time for RISC OS, the London Show is not to be missed. Just a few of the highlights:

-  The super-fast ARMiniX and PandaRO
-  Lots of hacks on the Raspberry Pi
-  RISC OS 5.2 for all machines
-  Impression X
-  The only working Phoebe prototype

Saturday 26 October 11am to 5pm

St Giles Hotel, Hounslow Road, Feltham, TW14 9AD

More information and the latest news on the show website:

<http://riscoslondonshow.co.uk>



Organised by the RISC OS User Group Of London

<http://rougol.jellybaby.net>



Abbey Press
Glastonbury

32 Norbins Road
BA6 9JG

www.archivemag.co.uk

(01458) 83 3603
iim@archivemag.co.uk